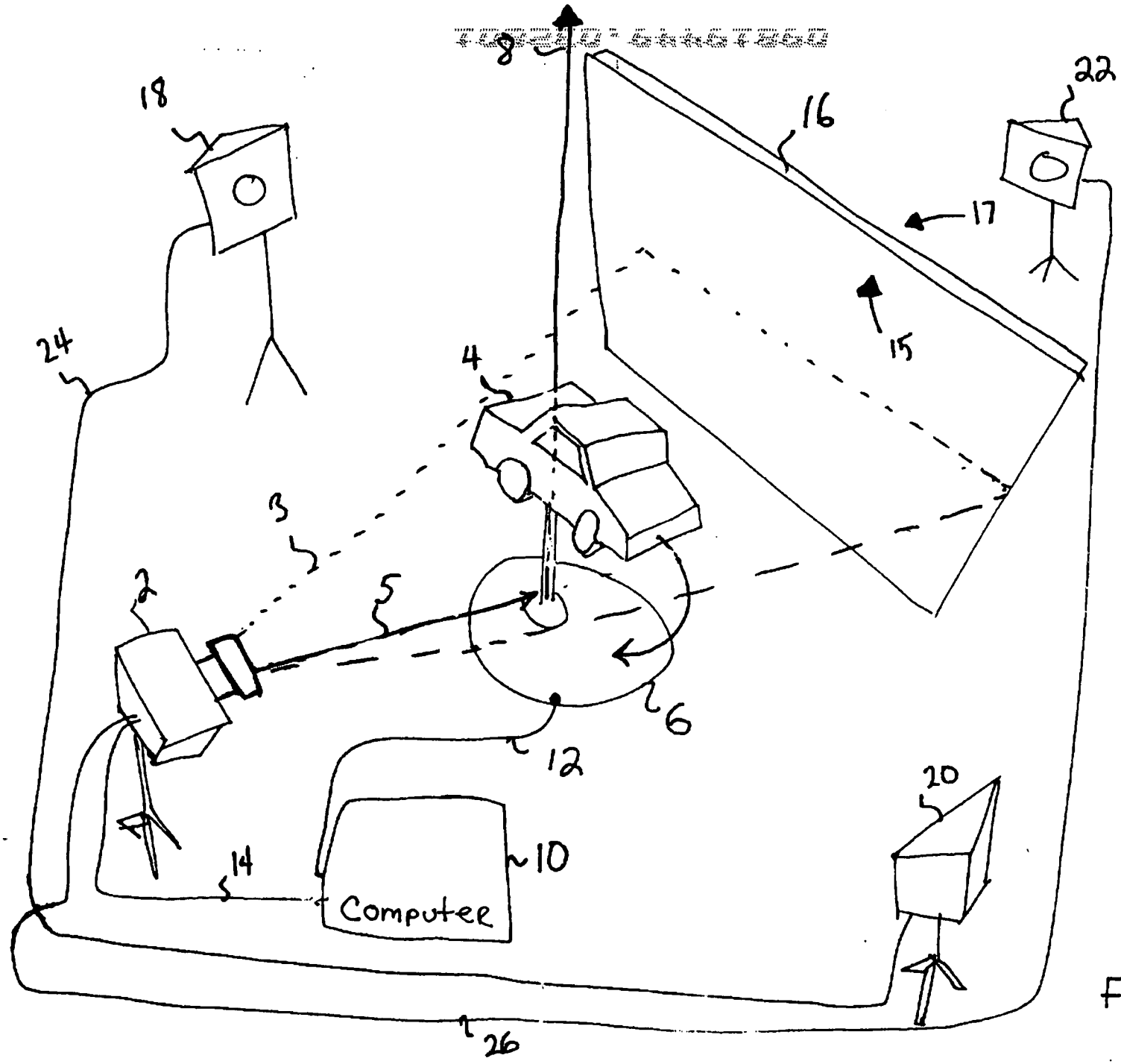


Fig. 1

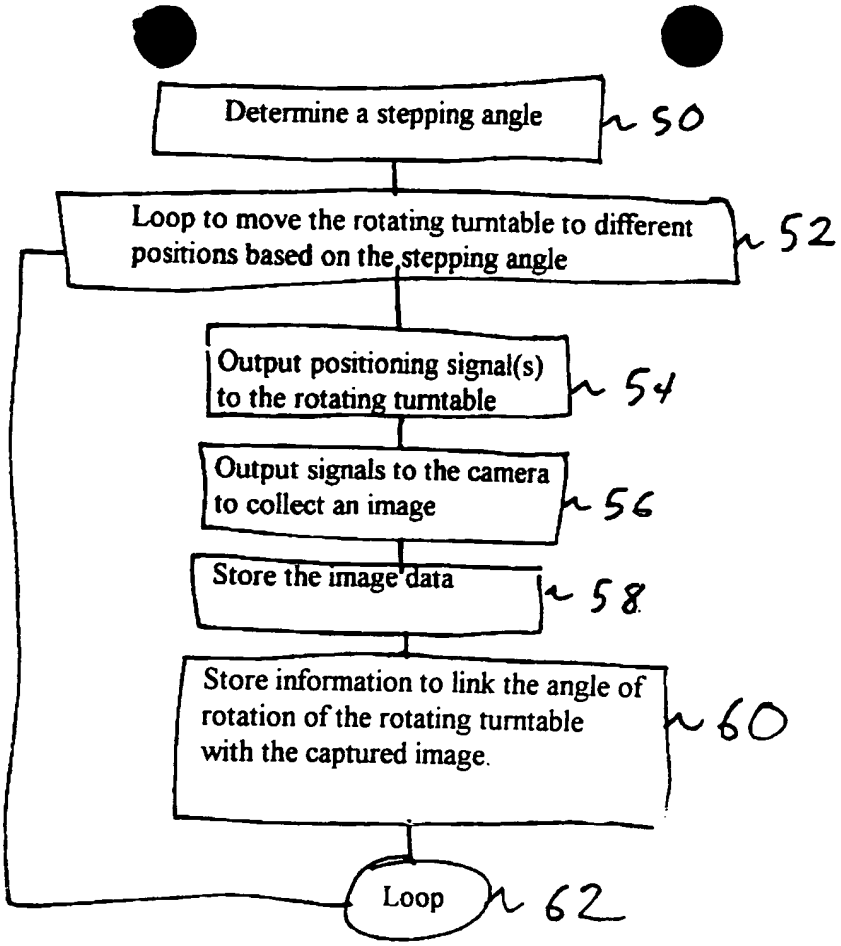

Determine a stepping angle
50
Loop to move the rotating turntable to different positions based on the stepping angle
52
Output positioning signal(s) to the rotating turntable
54
Output signals to the camera to collect an image
56
Store the image data
58
Store information to link the angle of rotation of the rotating turntable with the captured image.
60
Loop
62

Fig. 2

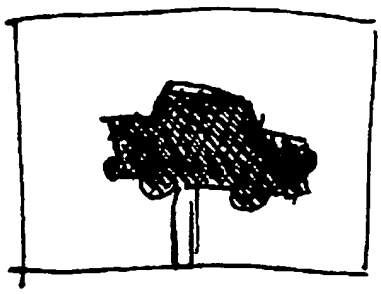


Fig. 3a

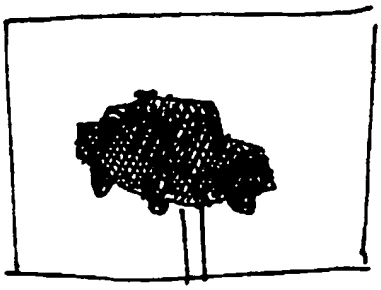


Fig. 3b

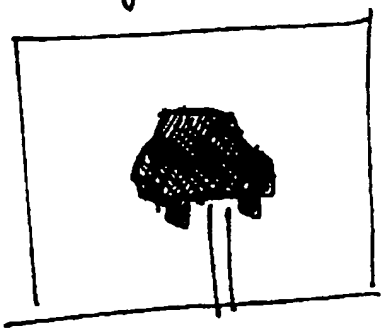


Fig. 3c

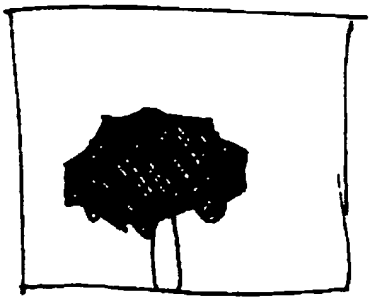


Fig. 3d

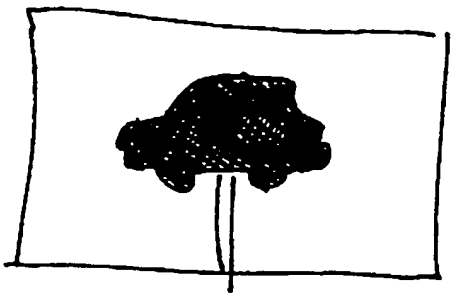


Fig. 3e

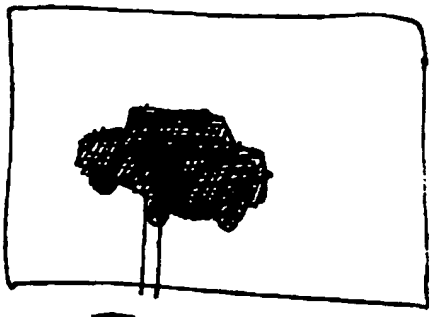


Fig. 3f

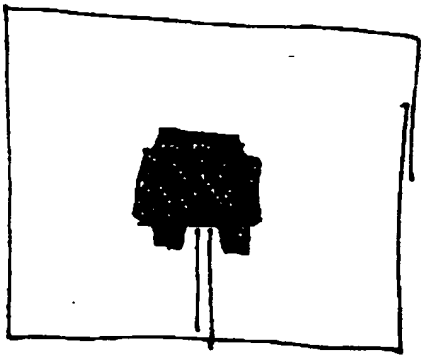


Fig. 3g

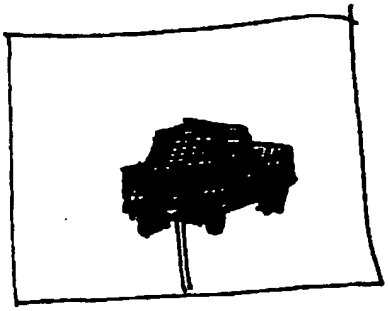


Fig. 3h

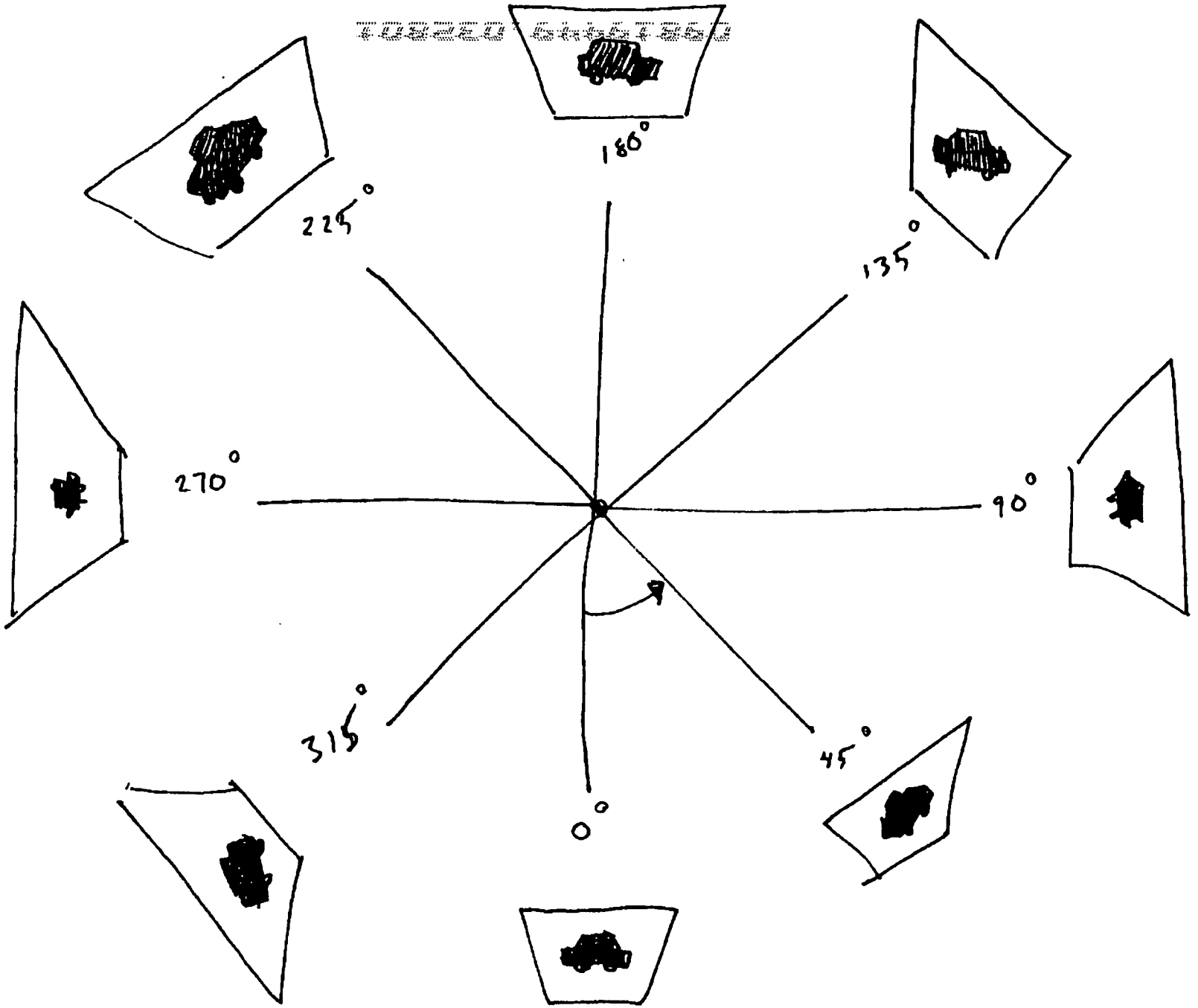


fig. 4

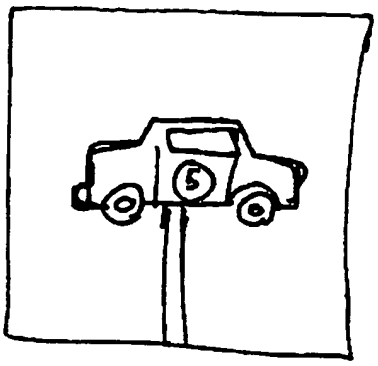


Fig. 5a

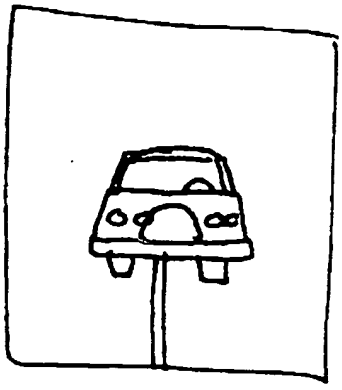


Fig. 5b

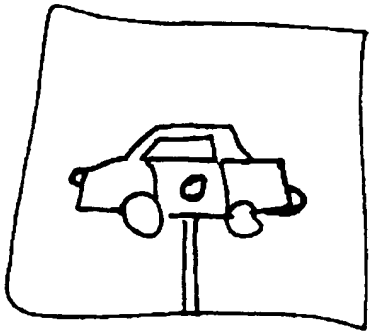


Fig. 5c

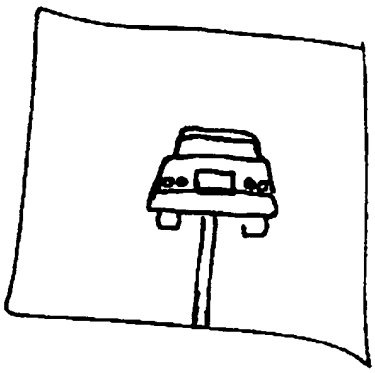


Fig. 5d

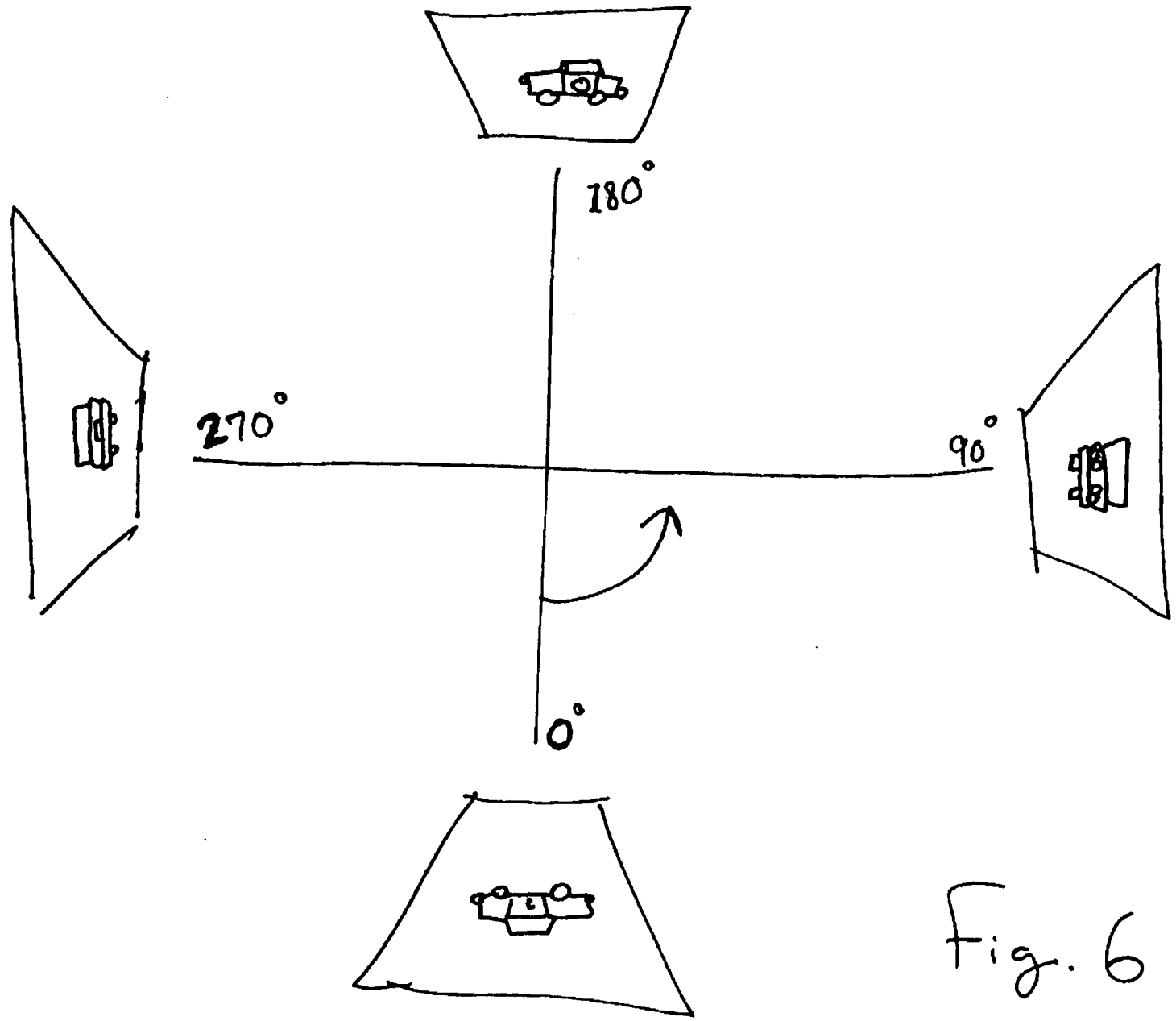


Fig. 6

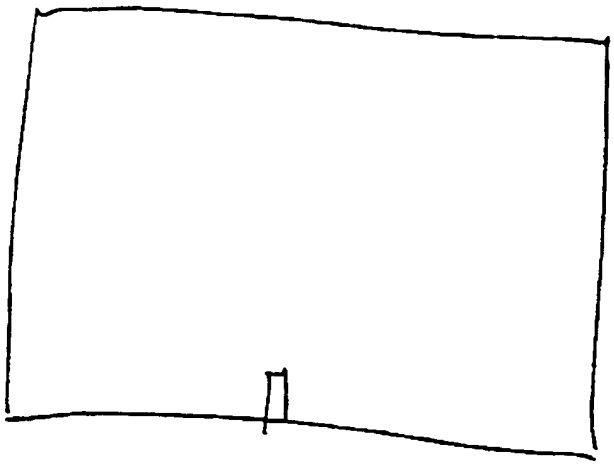


fig. 7

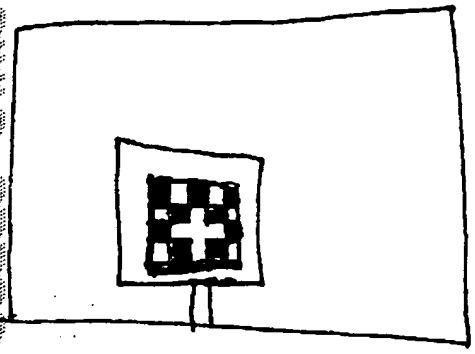


fig. 8a

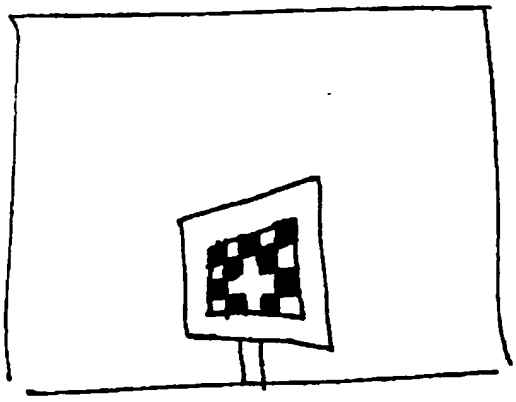


fig. 8b

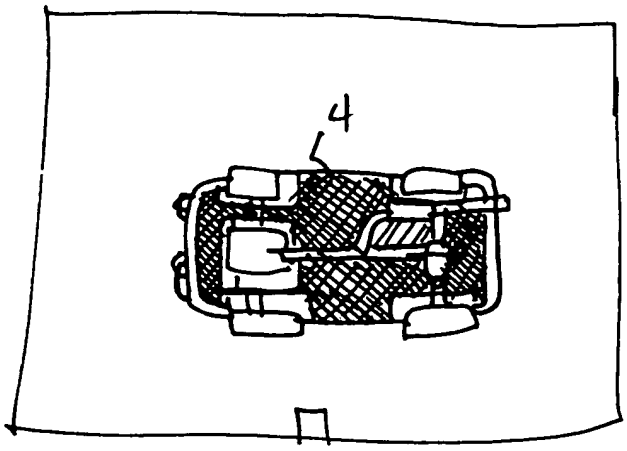


Fig. 9

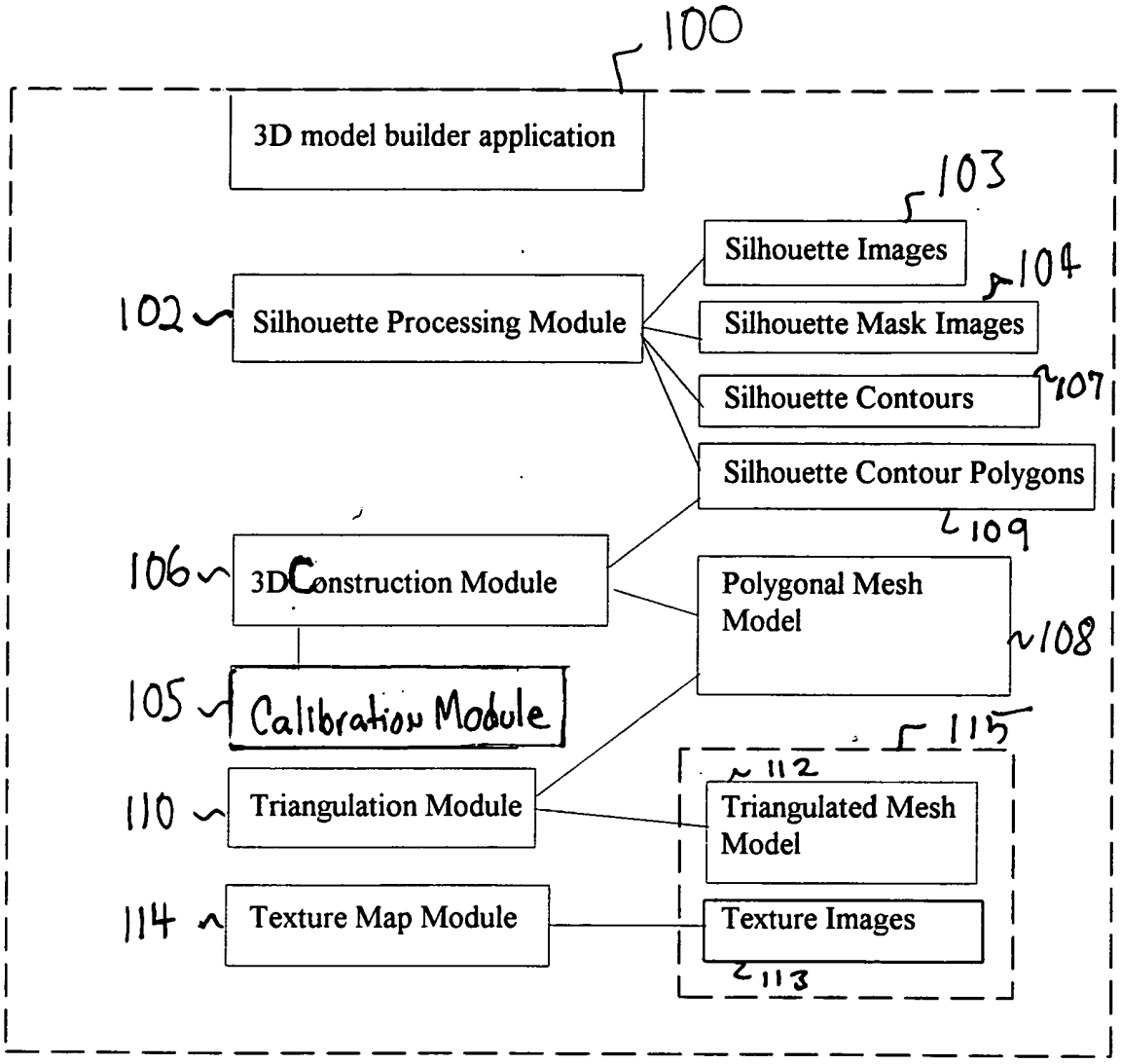

100
3D model builder application
103
Silhouette Images
104
Silhouette Mask Images
102
Silhouette Processing Module
107
Silhouette Contours
Silhouette Contour Polygons
109
106
3D Construction Module
Polygonal Mesh Model
108
105
Calibration Module
115
112
110
Triangulation Module
Triangulated Mesh Model
114
Texture Map Module
Texture Images
113

Fig. 10

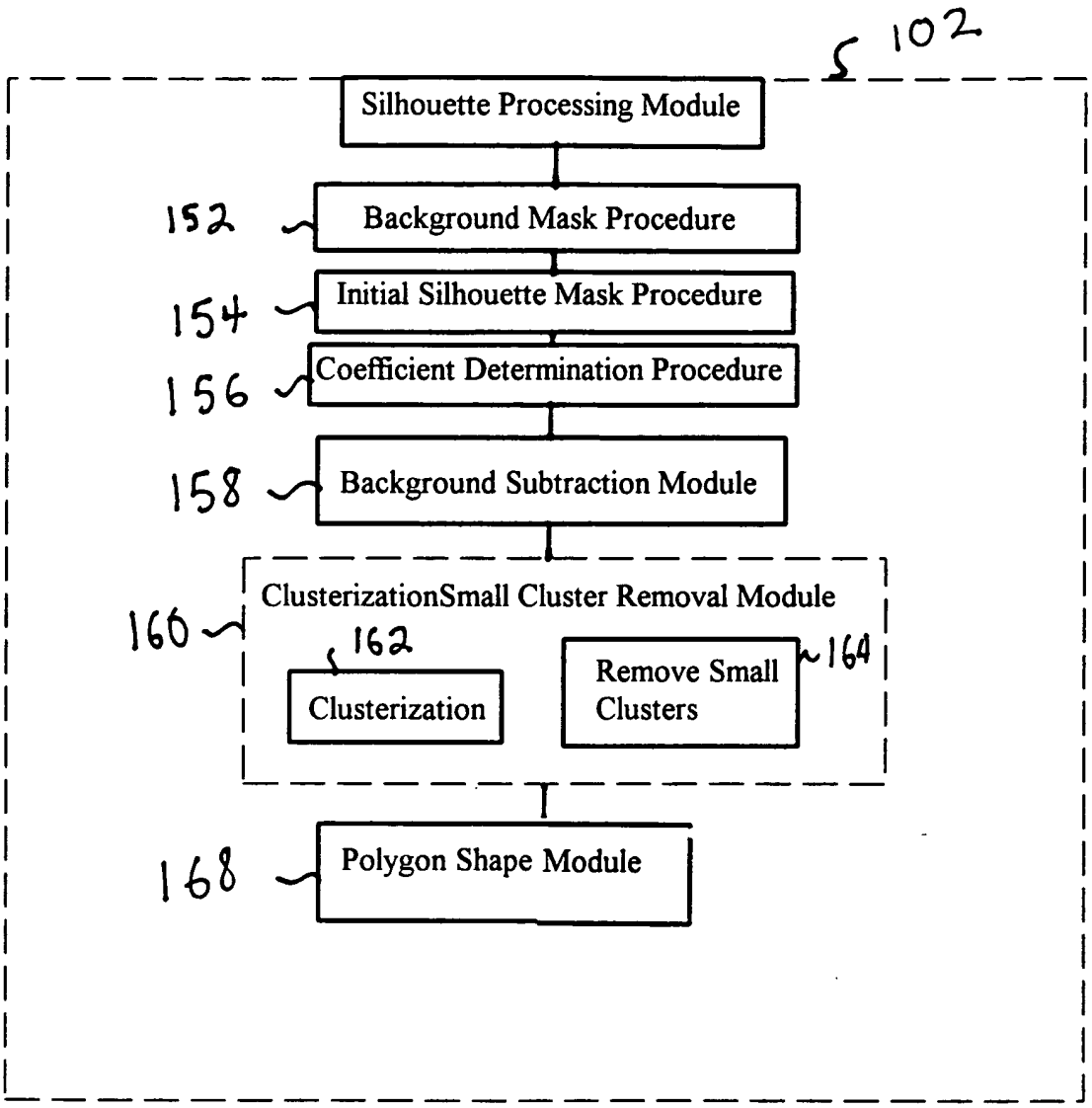

102
Silhouette Processing Module
152
Background Mask Procedure
154
Initial Silhouette Mask Procedure
156
Coefficient Determination Procedure
158
Background Subtraction Module
160
ClusterizationSmall Cluster Removal Module
162
Clusterization
164
Remove Small Clusters
168
Polygon Shape Module

Fig. 11

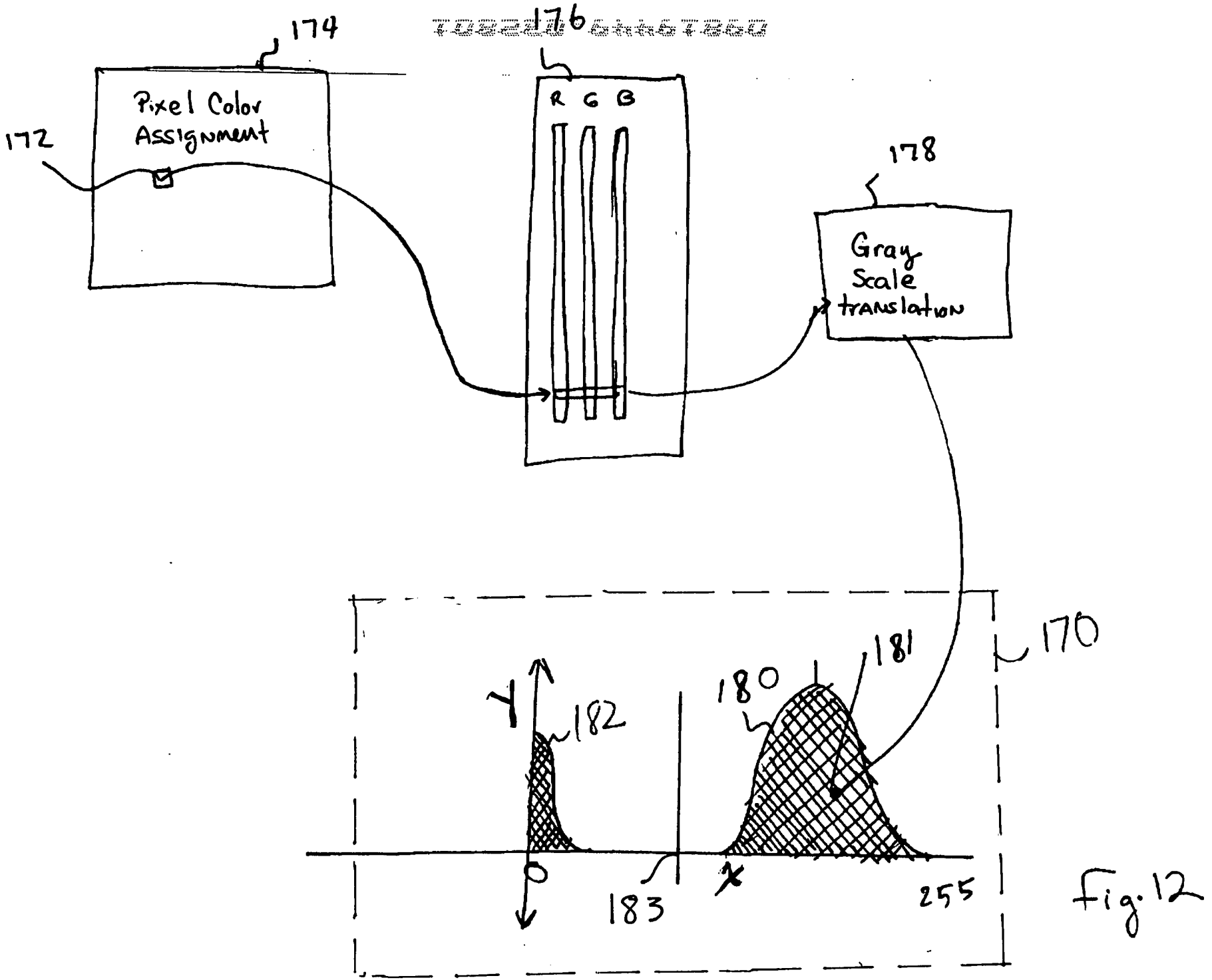

174
176
Pixel Color Assignment
172
R G B
178
Gray Scale translation
170
181
180
182
Y
0
183
X
255

fig. 12

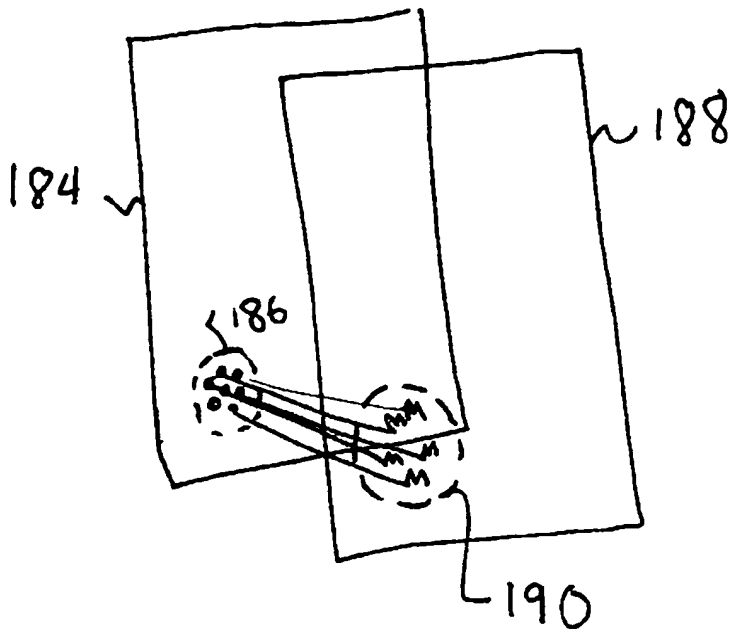


Fig. 13

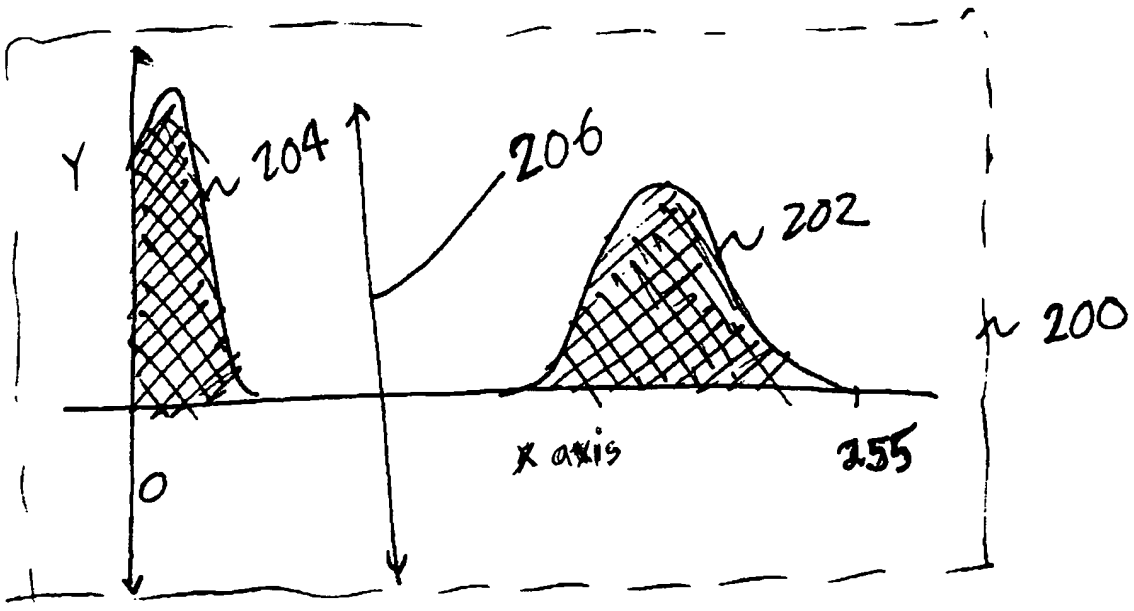


Fig. 14

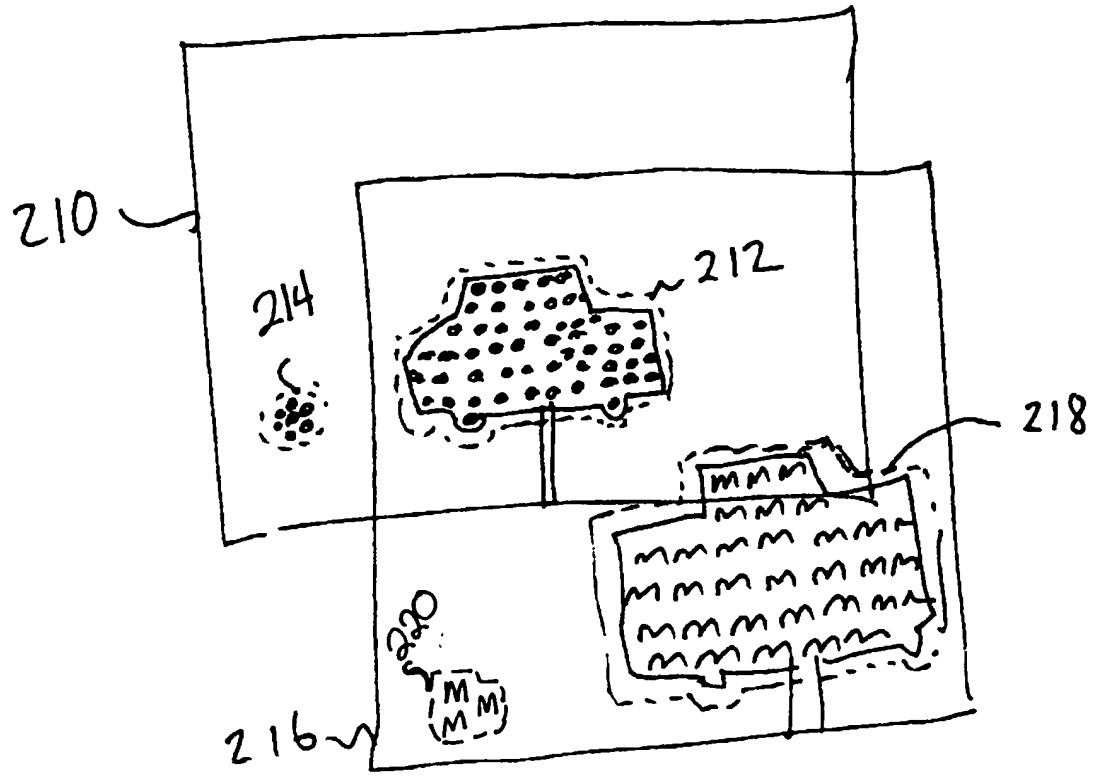

210
212
214
218
220
216

Fig. 15

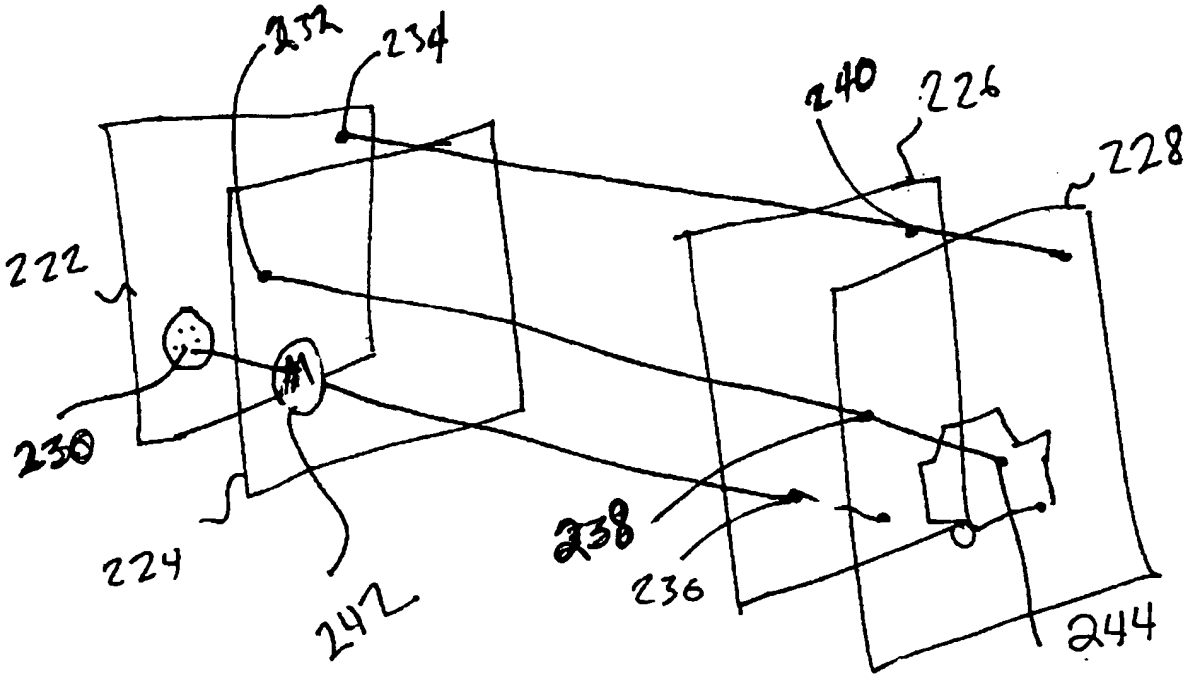

232
234
240
226
228
222
230
#1
224
242
238
236
244

Fig. 16

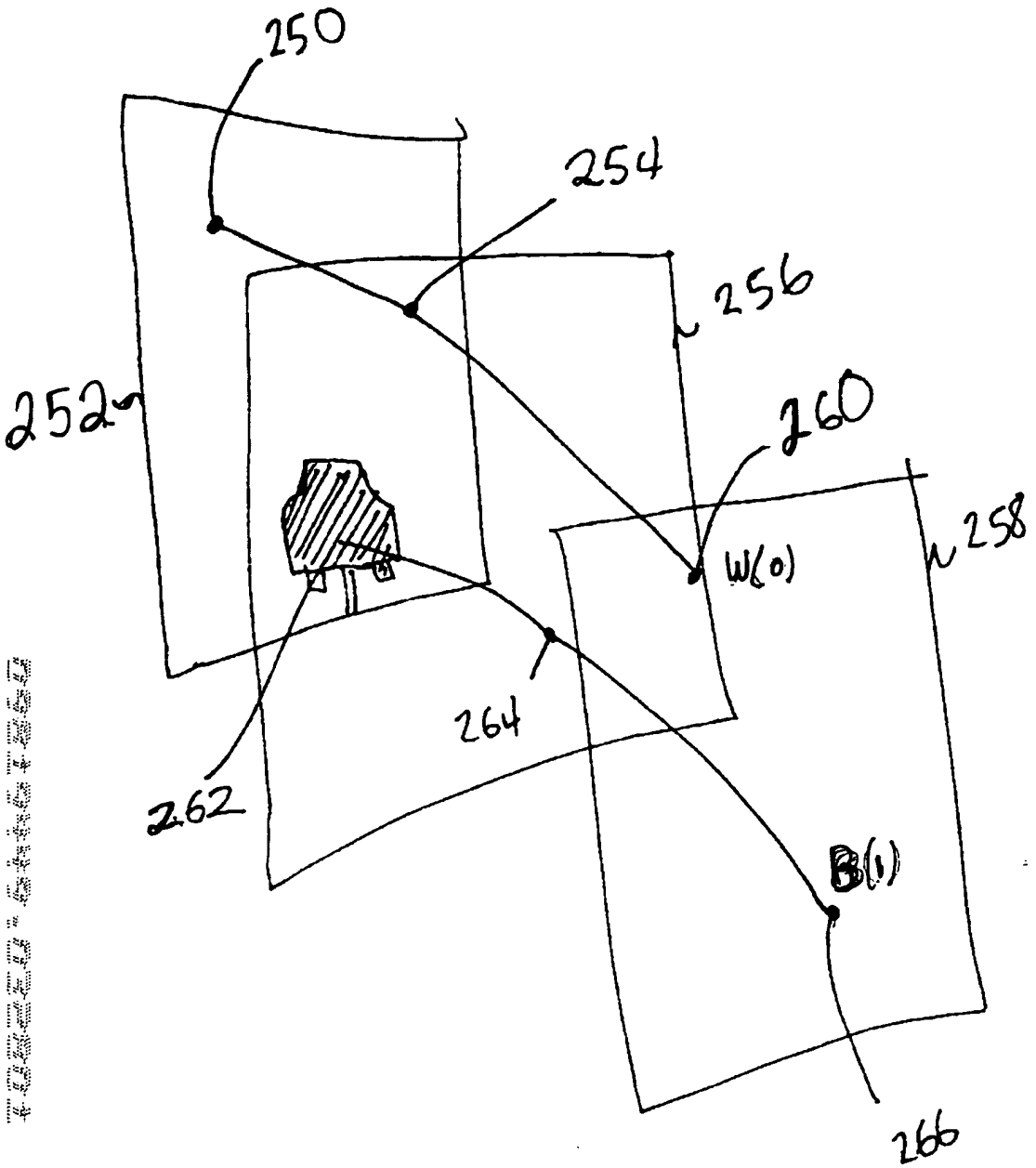


fig 17

Fig. 18

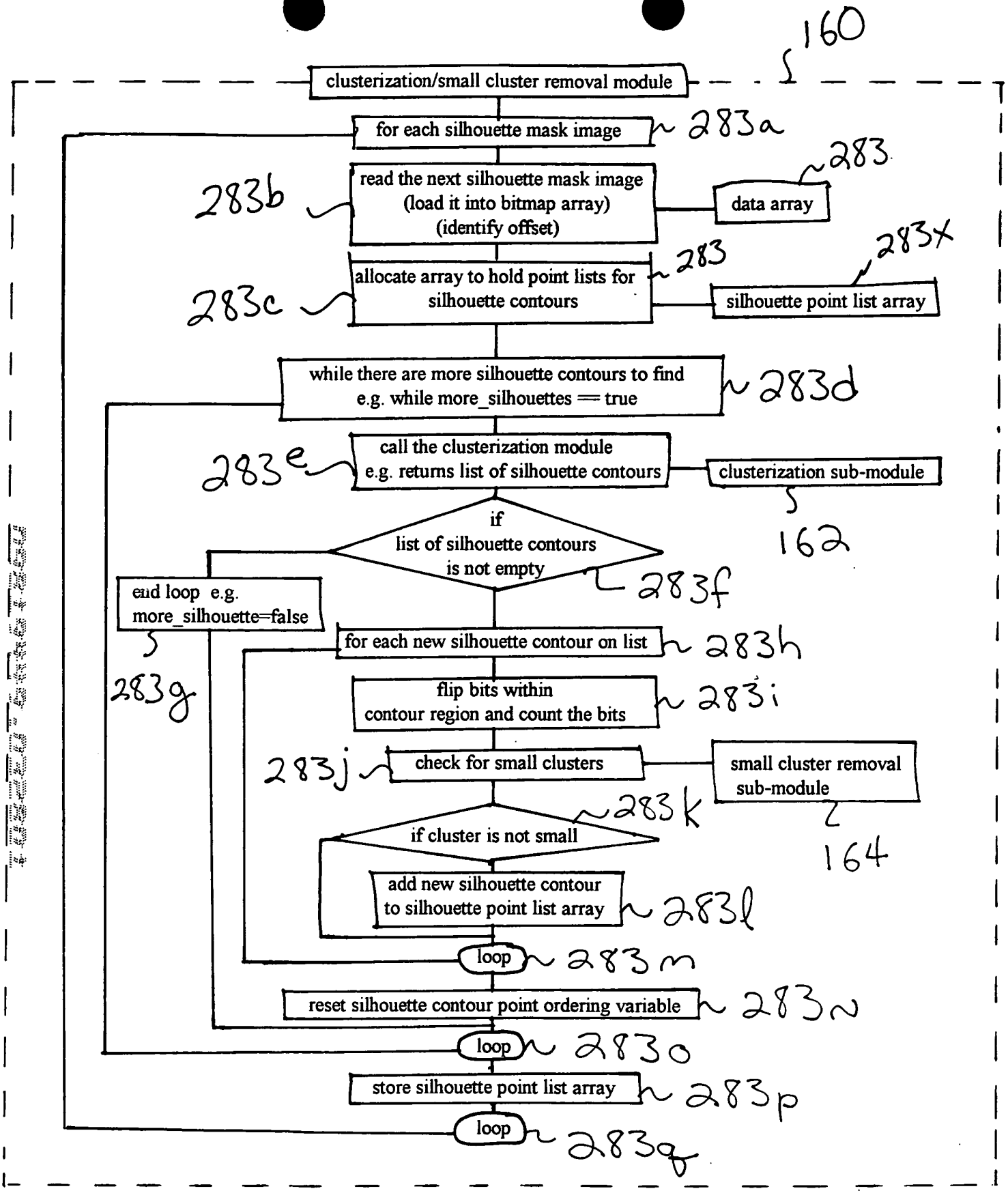

160
clusterization/small cluster removal module
for each silhouette mask image
283a
283b
read the next silhouette mask image
(load it into bitmap array)
(identify offset)
283
data array
283x
283c
allocate array to hold point lists for
silhouette contours
silhouette point list array
while there are more silhouette contours to find
e.g. while more_silhouettes = true
283d
283e
call the clusterization module
e.g. returns list of silhouette contours
clusterization sub-module
162
if
list of silhouette contours
is not empty
283f
end loop e.g.
more_silhouette=false
283g
for each new silhouette contour on list
283h
flip bits within
contour region and count the bits
283i
283j
check for small clusters
small cluster removal
sub-module
164
283k
if cluster is not small
add new silhouette contour
to silhouette point list array
283l
loop
283m
reset silhouette contour point ordering variable
283n
loop
283o
store silhouette point list array
283p
loop
283q

Fig. 19a

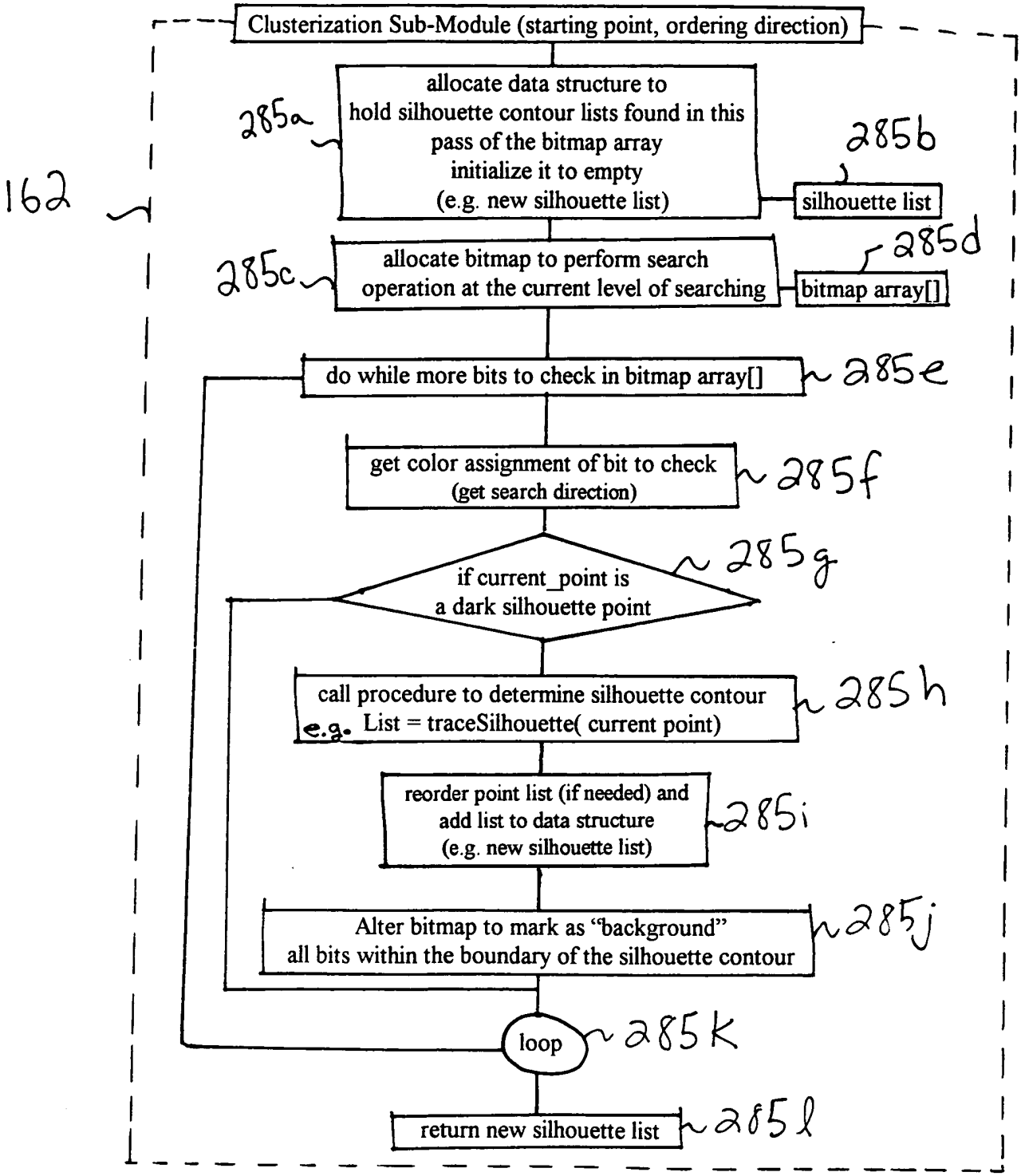

Clusterization Sub-Module (starting point, ordering direction)
162
allocate data structure to hold silhouette contour lists found in this pass of the bitmap array initialize it to empty (e.g. new silhouette list)
285a
285b
silhouette list
allocate bitmap to perform search operation at the current level of searching
285c
285d
bitmap array[]
do while more bits to check in bitmap array[]
285e
get color assignment of bit to check (get search direction)
285f
if current_point is a dark silhouette point
285g
call procedure to determine silhouette contour
e.g. List = traceSilhouette(current point)
285h
reorder point list (if needed) and add list to data structure (e.g. new silhouette list)
285i
Alter bitmap to mark as "background" all bits within the boundary of the silhouette contour
285j
loop
285k
return new silhouette list
285l

Fig. 19b

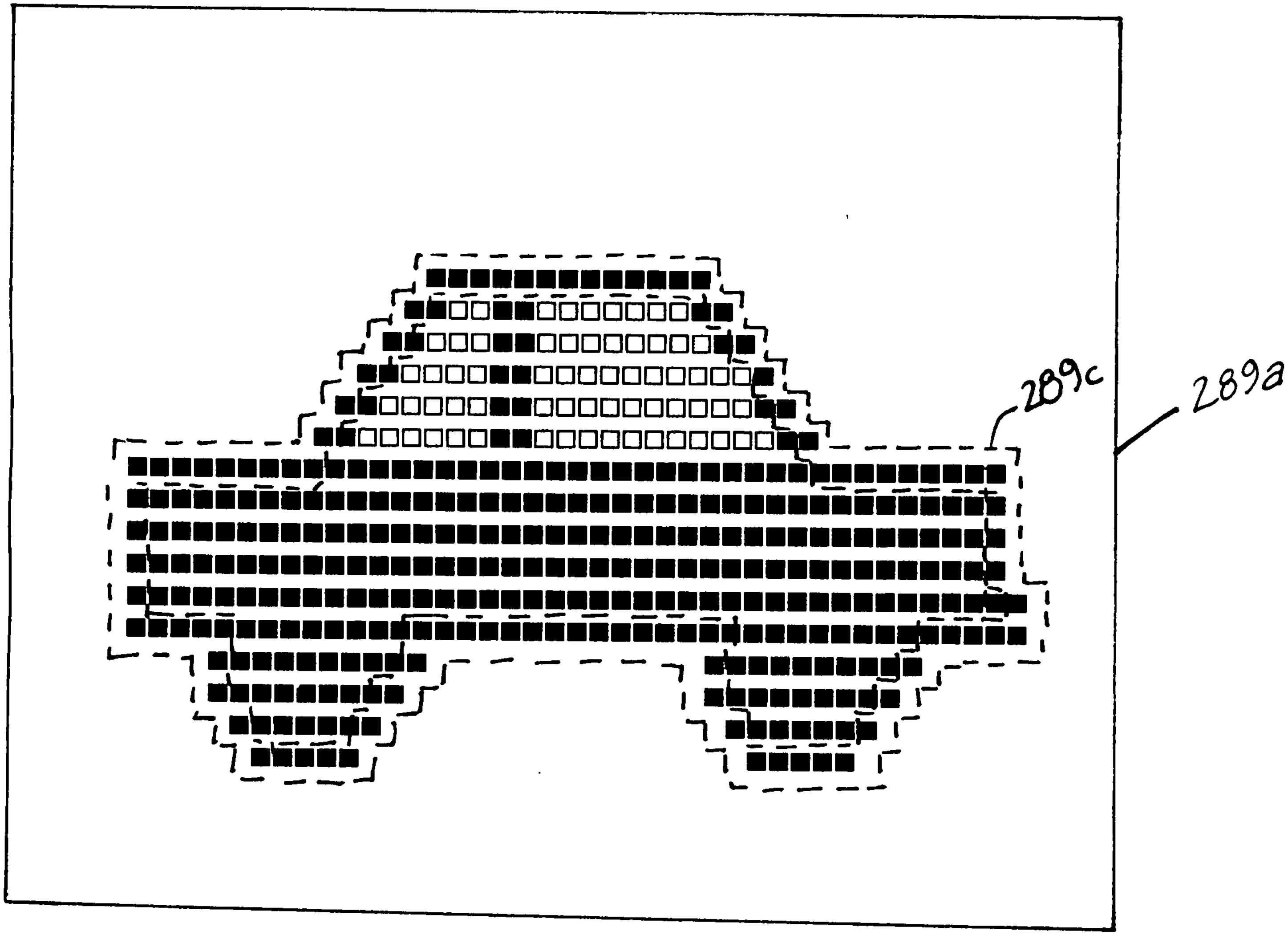


Fig. 19c

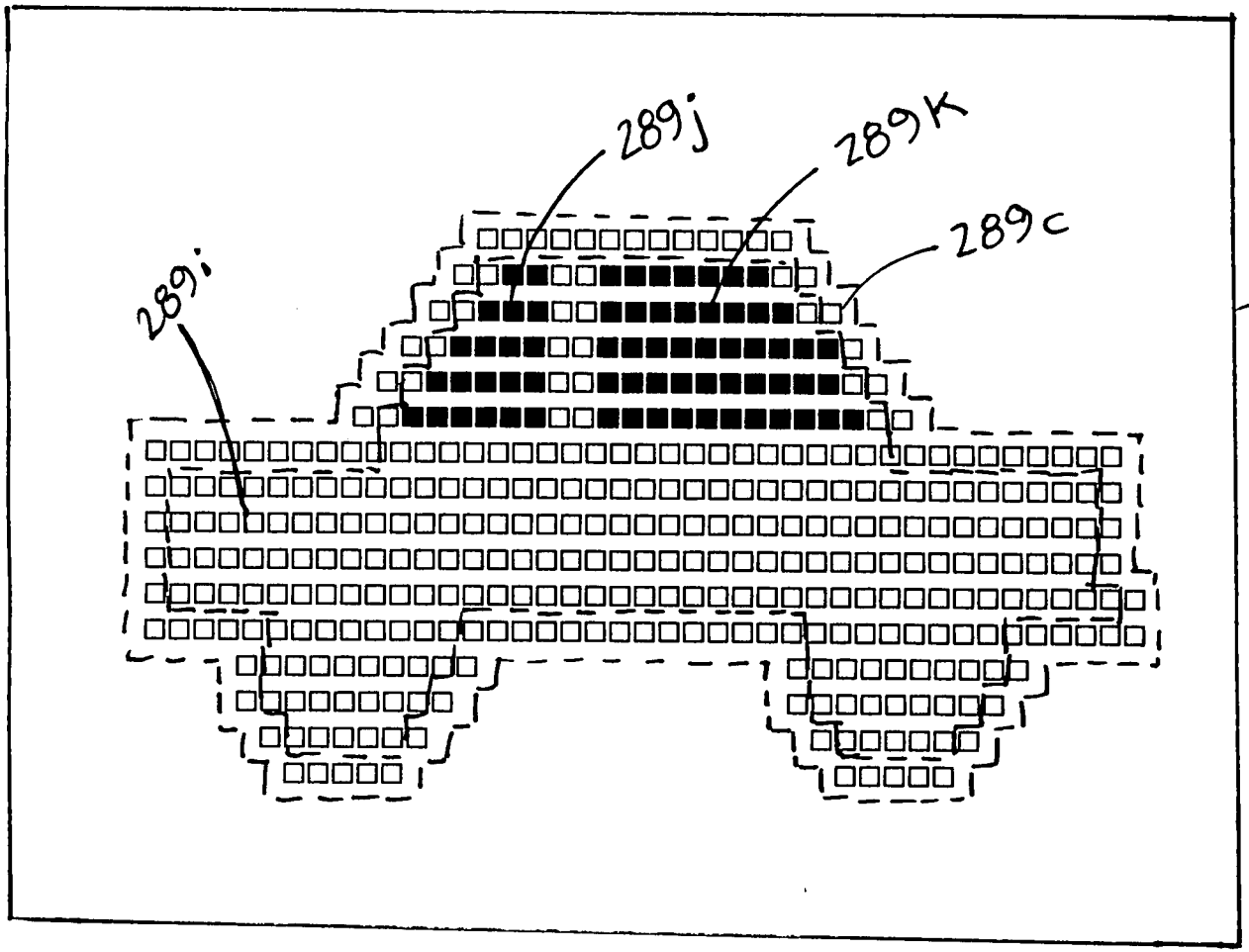


Fig. 19d

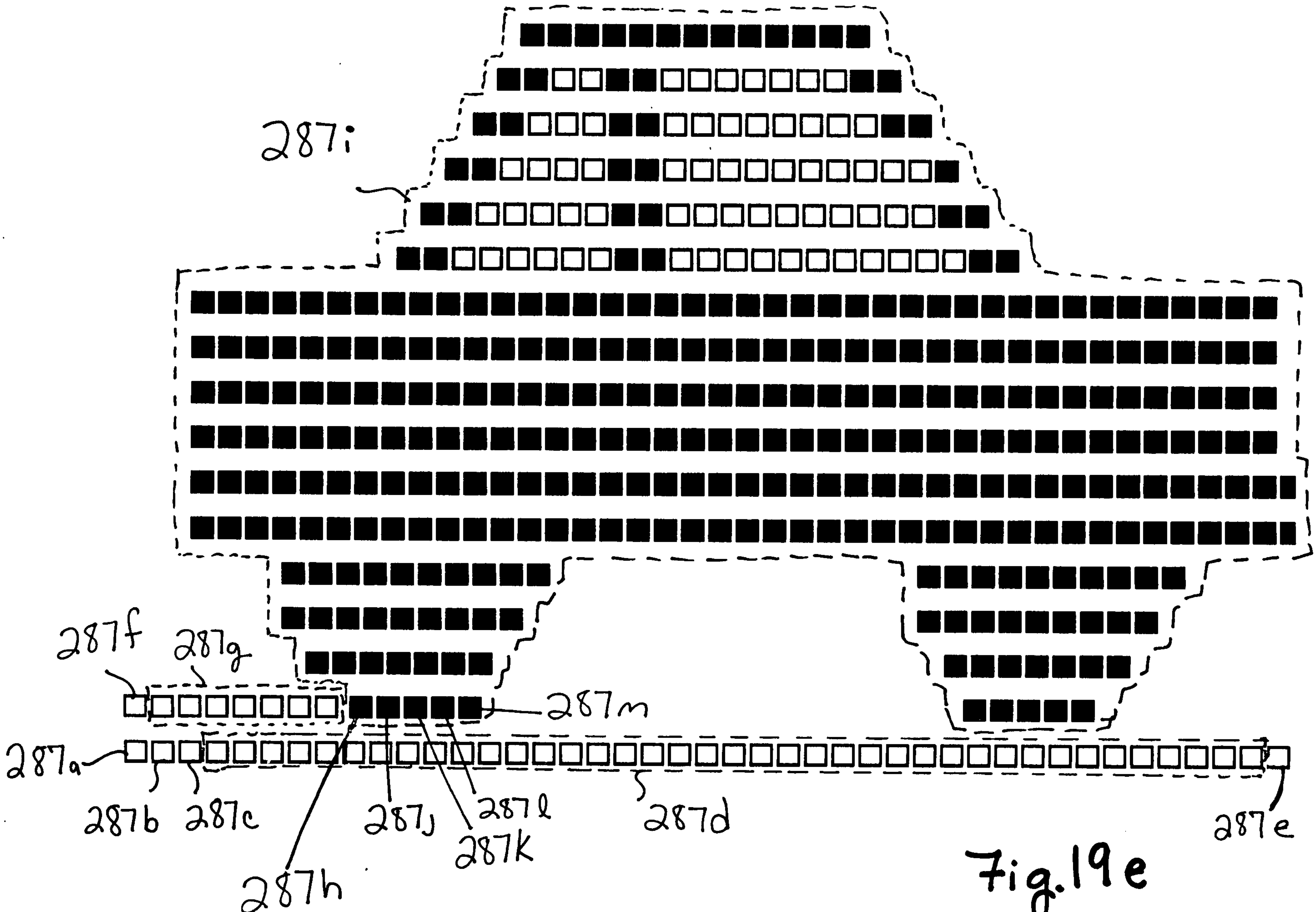

287i
287f
287g
287m
287a
287b
287c
287j
287l
287k
287h
287d
287e

Fig.19e

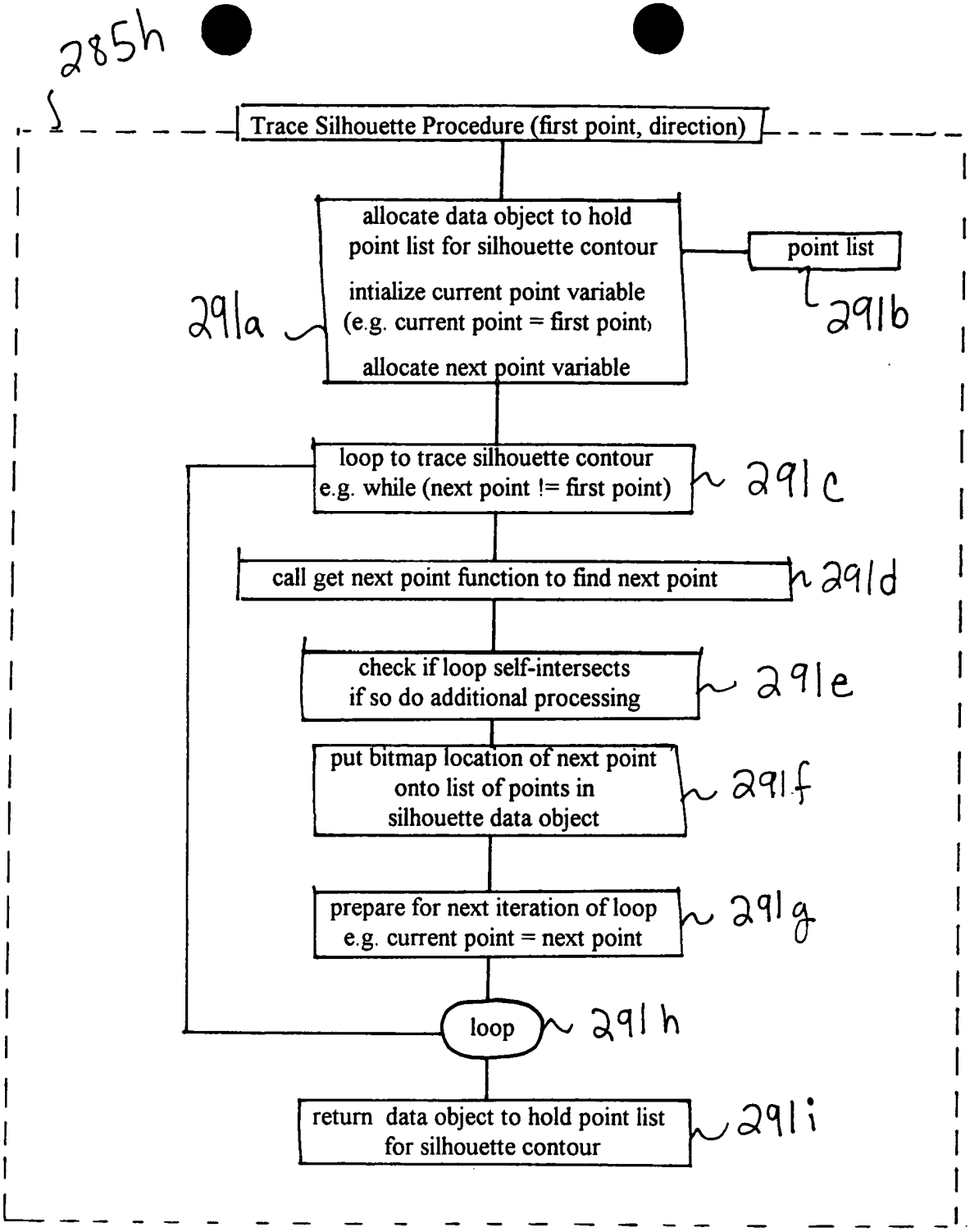

285h
Trace Silhouette Procedure (first point, direction)
allocate data object to hold point list for silhouette contour
intialize current point variable (e.g. current point = first point)
allocate next point variable
point list
291a
291b
loop to trace silhouette contour e.g. while (next point != first point)
291c
call get next point function to find next point
291d
check if loop self-intersects if so do additional processing
291e
put bitmap location of next point onto list of points in silhouette data object
291f
prepare for next iteration of loop e.g. current point = next point
291g
loop
291h
return data object to hold point list for silhouette contour
291i

Fig. 20a

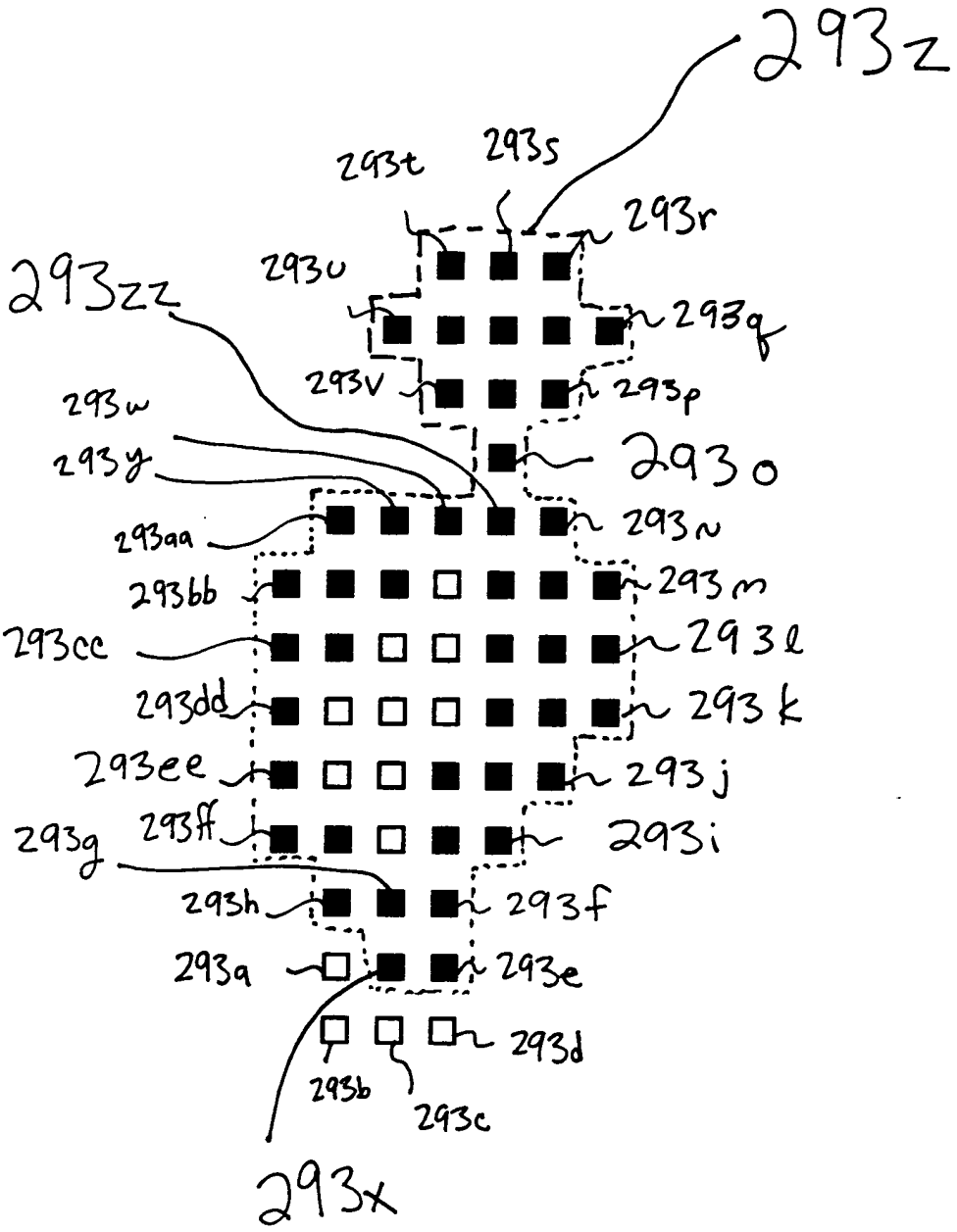

293z
293t
293s
293r
293u
293zz
293q
293v
293p
293w
293o
293y
293aa
293n
293bb
293m
293cc
293l
293dd
293k
293ee
293j
293g
293ff
293i
293h
293f
293a
293e
293d
293b
293c
293x

Fig. 20b

Fig. 21 a

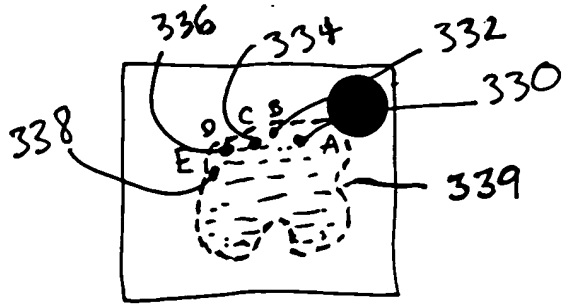


Fig. 21 b

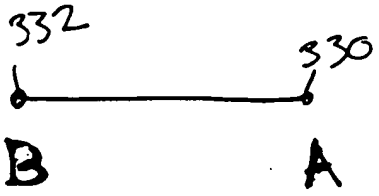


Fig. 21 c

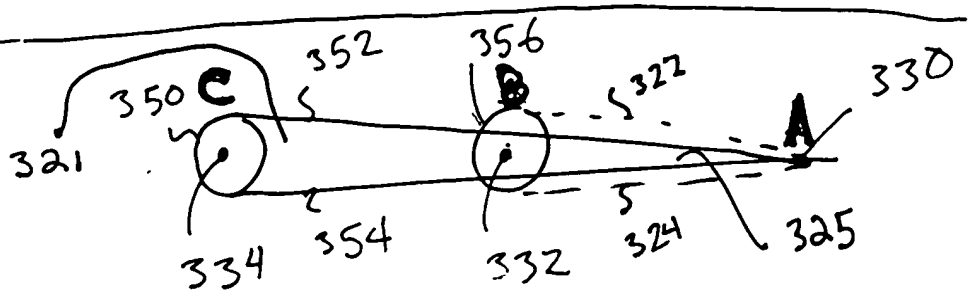


Fig. 21 d

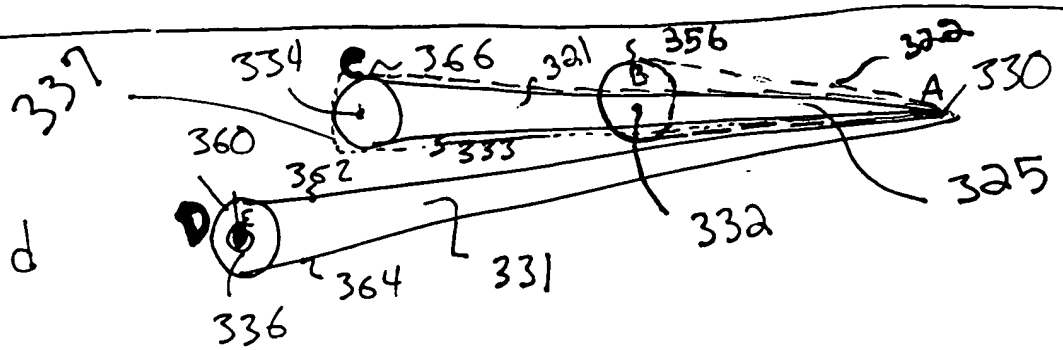
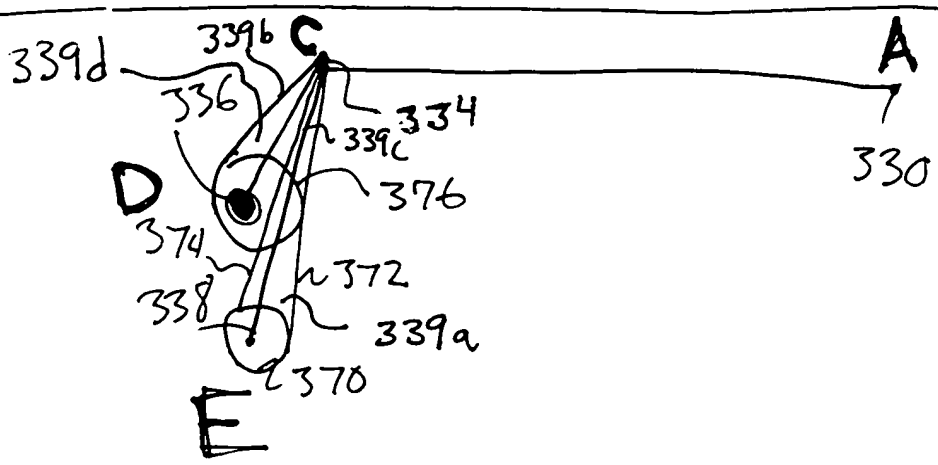


Fig. 21 e



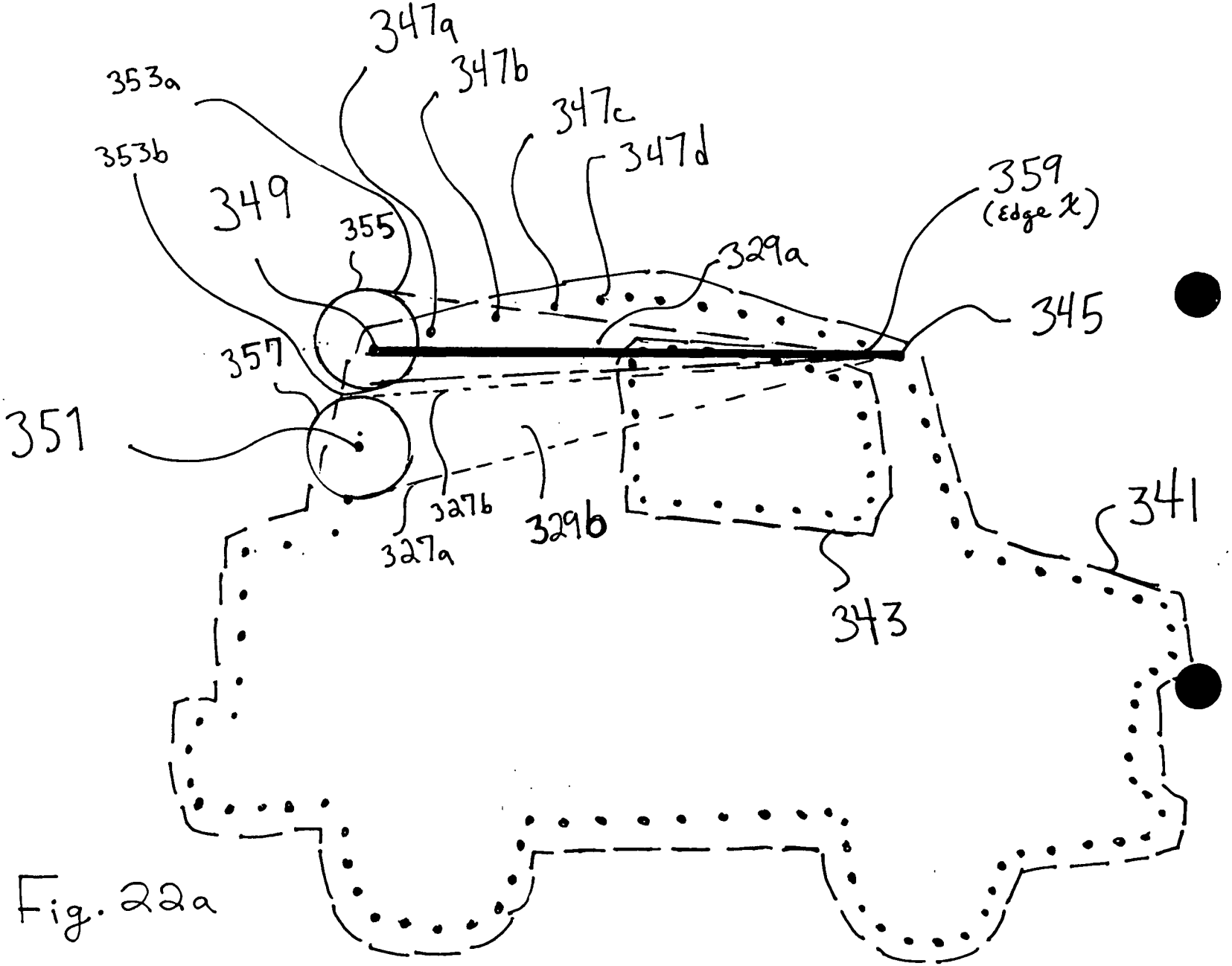


Fig. 22a

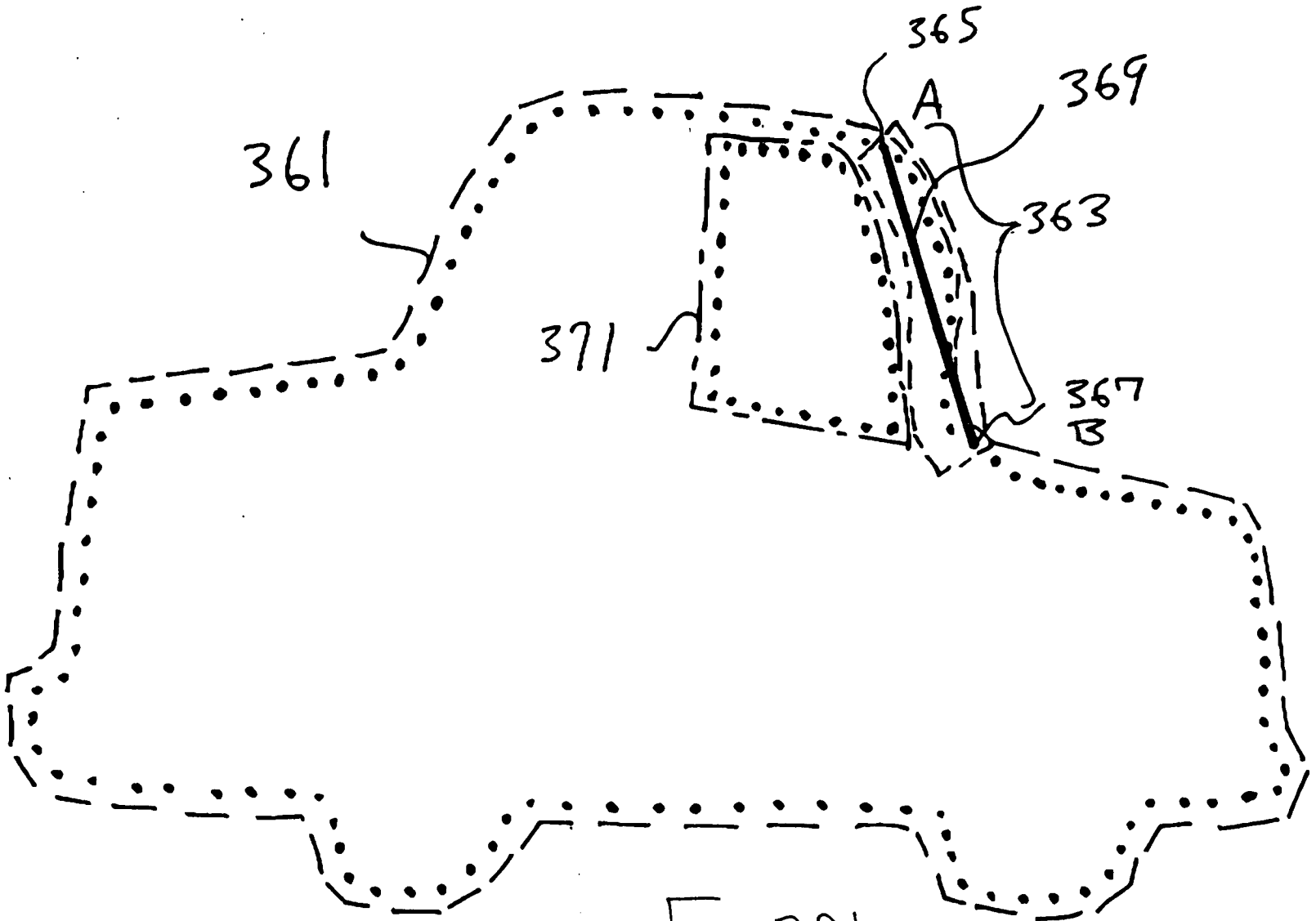


Fig. 22b

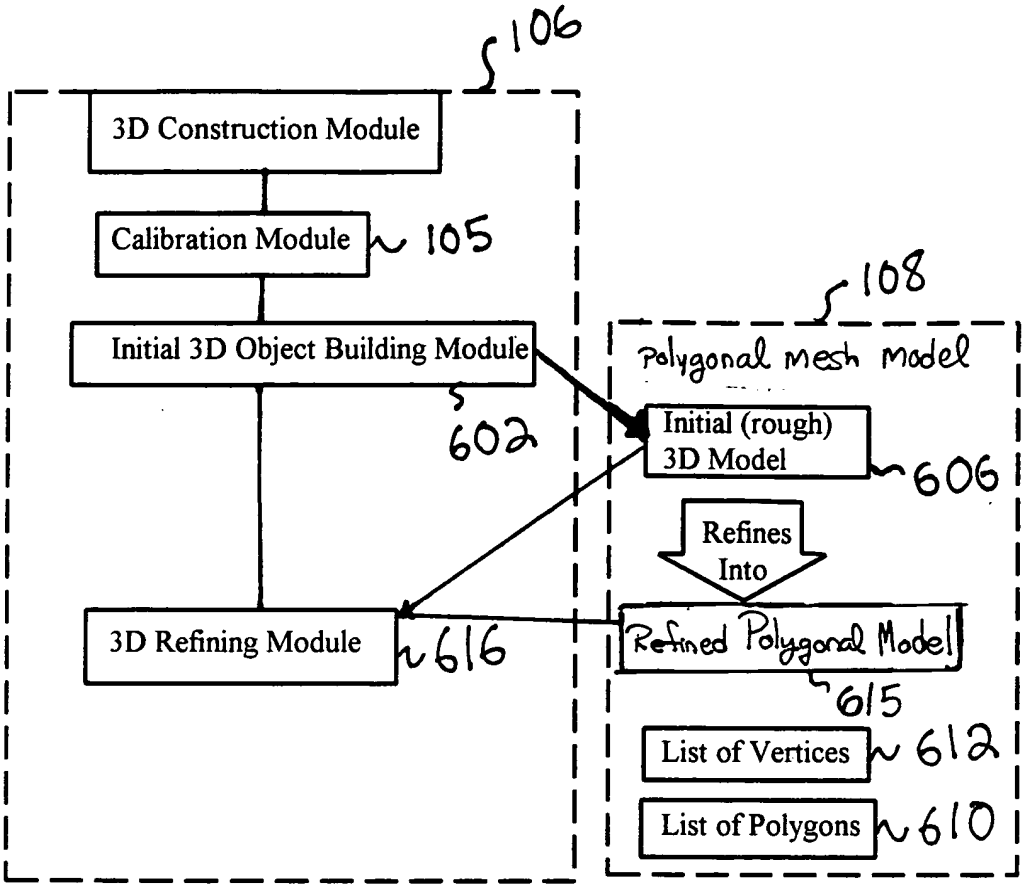

106
3D Construction Module
Calibration Module
105
Initial 3D Object Building Module
602
3D Refining Module
616
108
Polygonal Mesh Model
Initial (rough) 3D Model
606
Refines Into
Refined Polygonal Model
615
List of Vertices
612
List of Polygons
610

Fig. 23

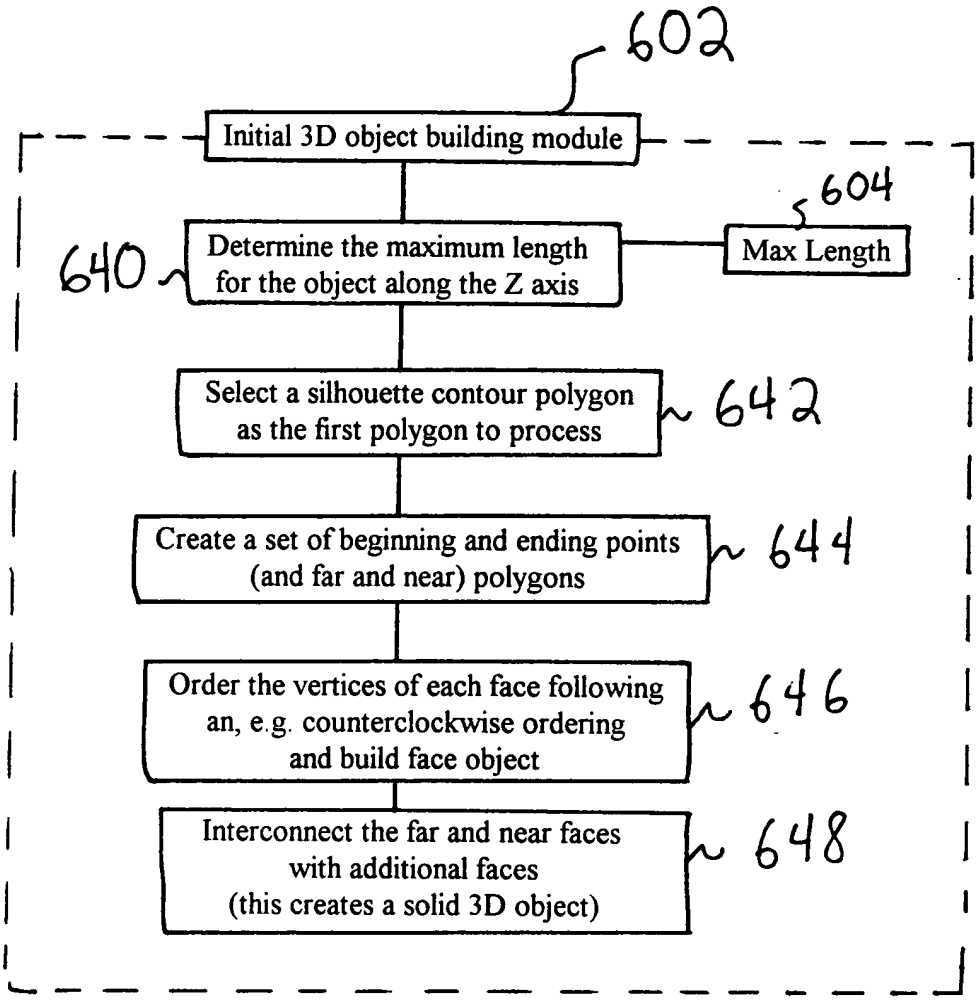

602
Initial 3D object building module
604
Max Length
640
Determine the maximum length for the object along the Z axis
Select a silhouette contour polygon as the first polygon to process
642
Create a set of beginning and ending points (and far and near) polygons
644
Order the vertices of each face following an, e.g. counterclockwise ordering and build face object
646
Interconnect the far and near faces with additional faces (this creates a solid 3D object)
648

Fig. 24

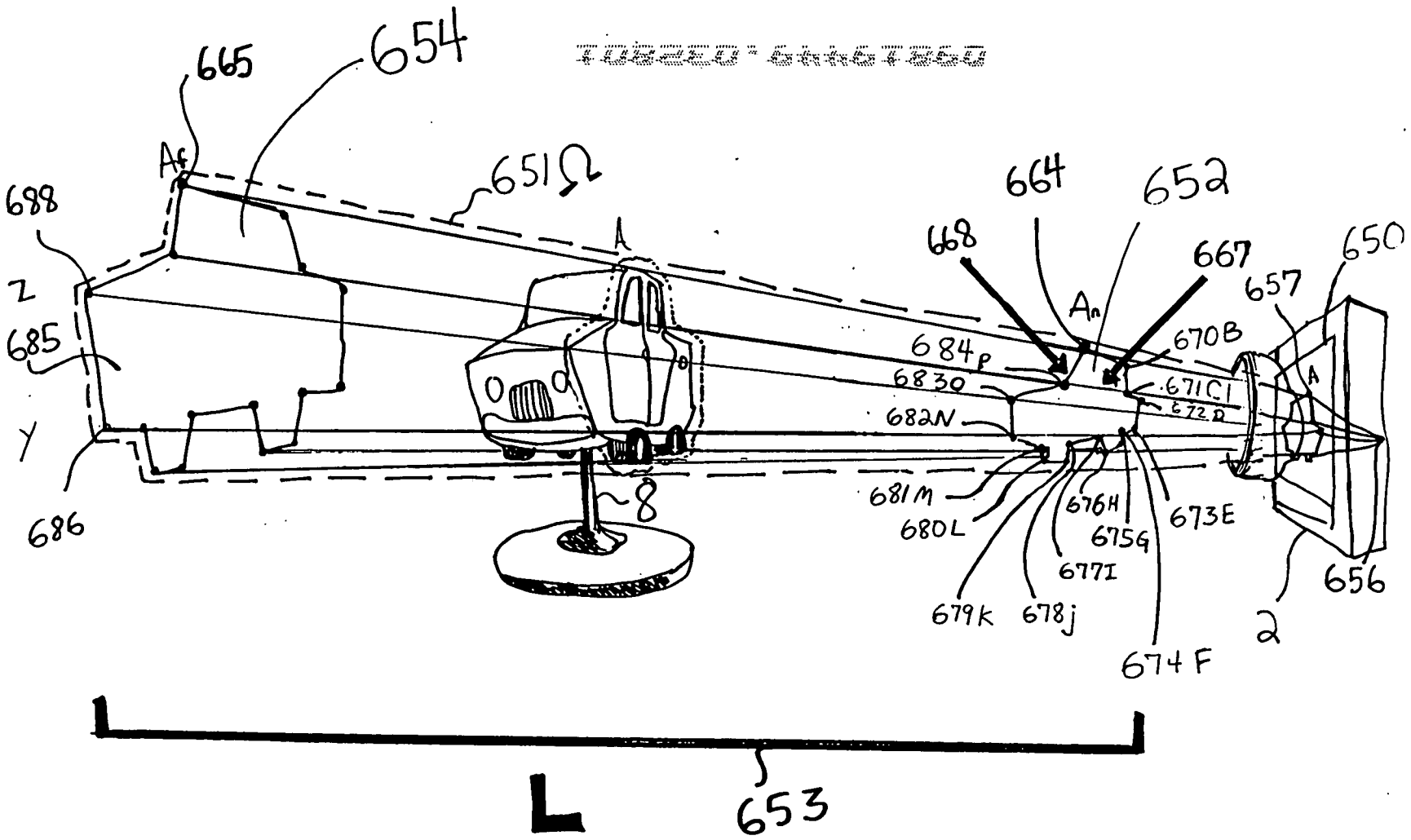

665
654
651Ω
664
652
650
688
668
667
657
Z
685
684p
670B
671C1
672D
682N
Y
686
681M
680L
679K
678j
677I
676H
675G
674F
673E
656
653
L

Fig. 25

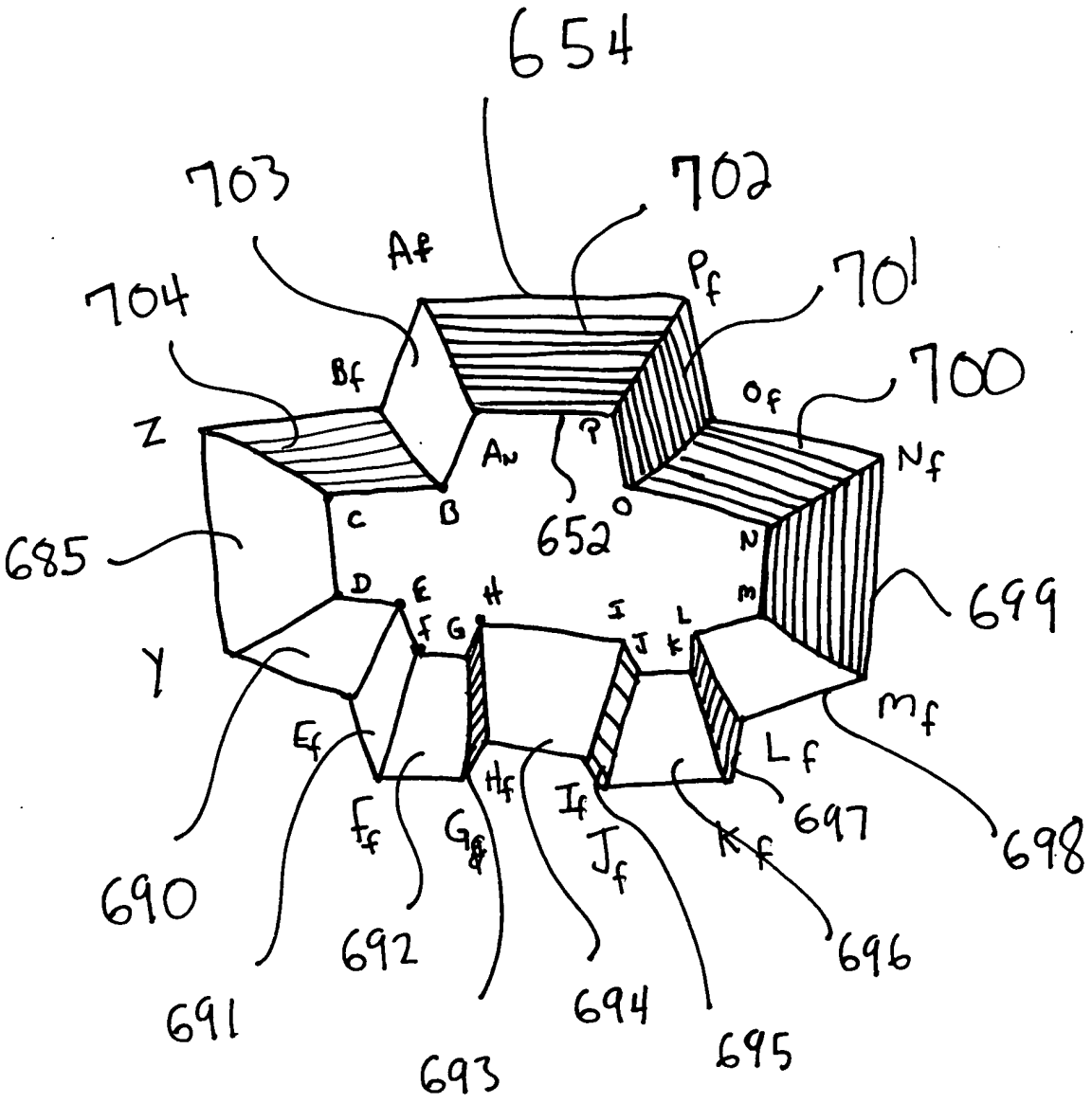

654
703
702
704
Af
Pf
701
Bf
Of
700
Z
Aw
P
Nf
C
B
652
O
685
N
D
E
H
I
L
699
m
f
G
J
k
Y
mf
Ef
Lf
Hf
If
Ff
Gf
Jf
Kf
697
698
690
692
696
691
694
693
695

Fig. 26

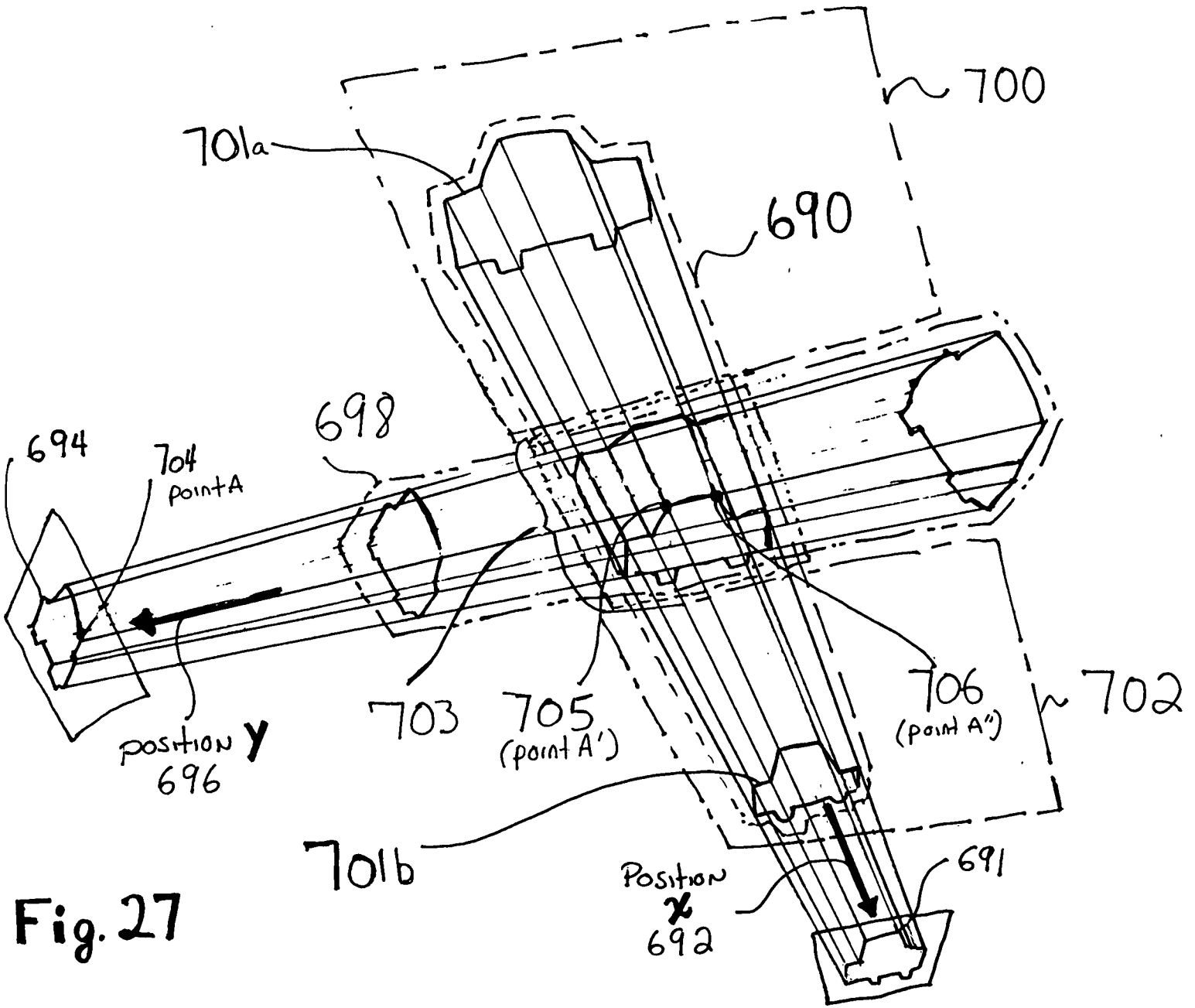


Fig. 27

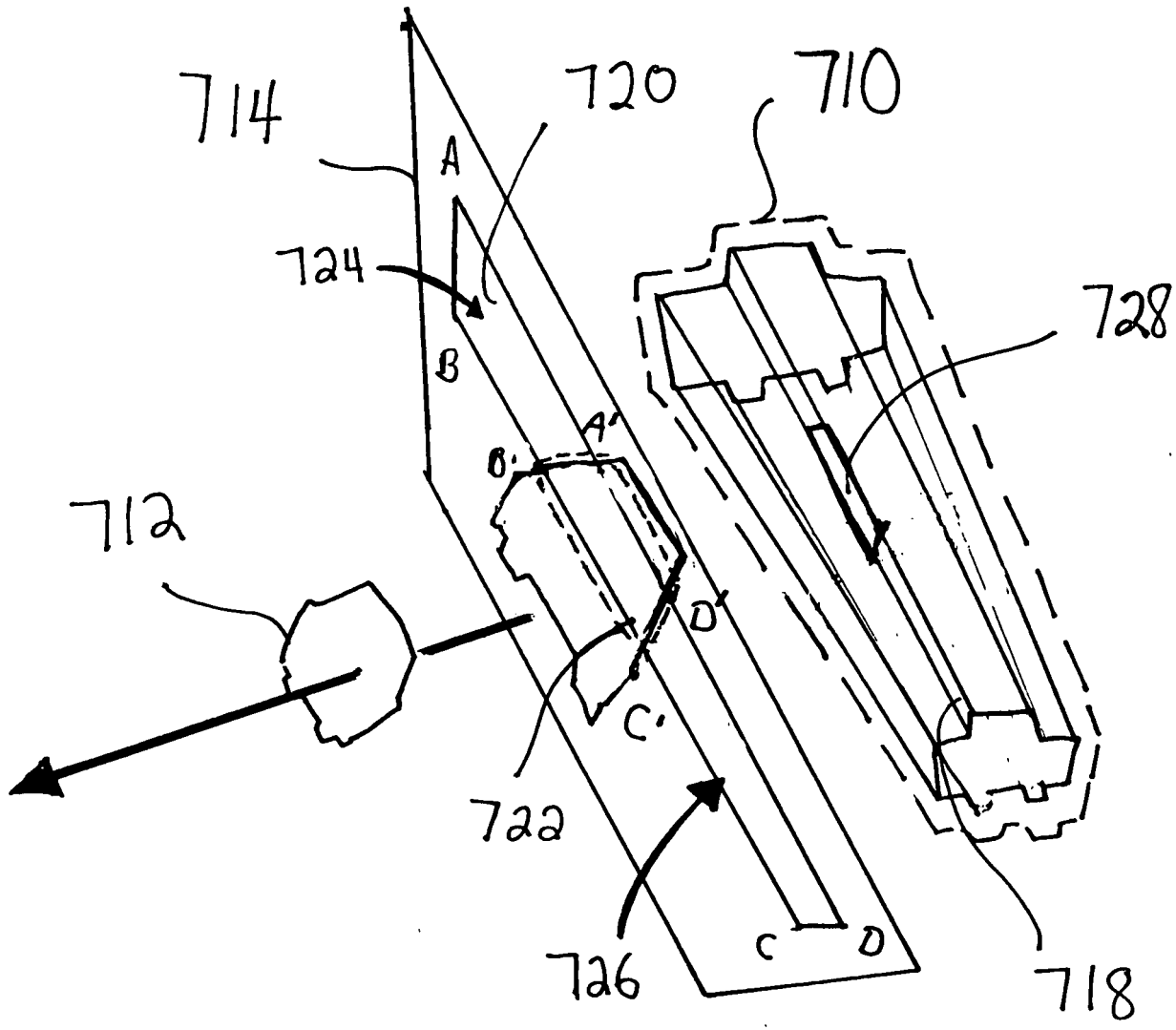


Fig. 28

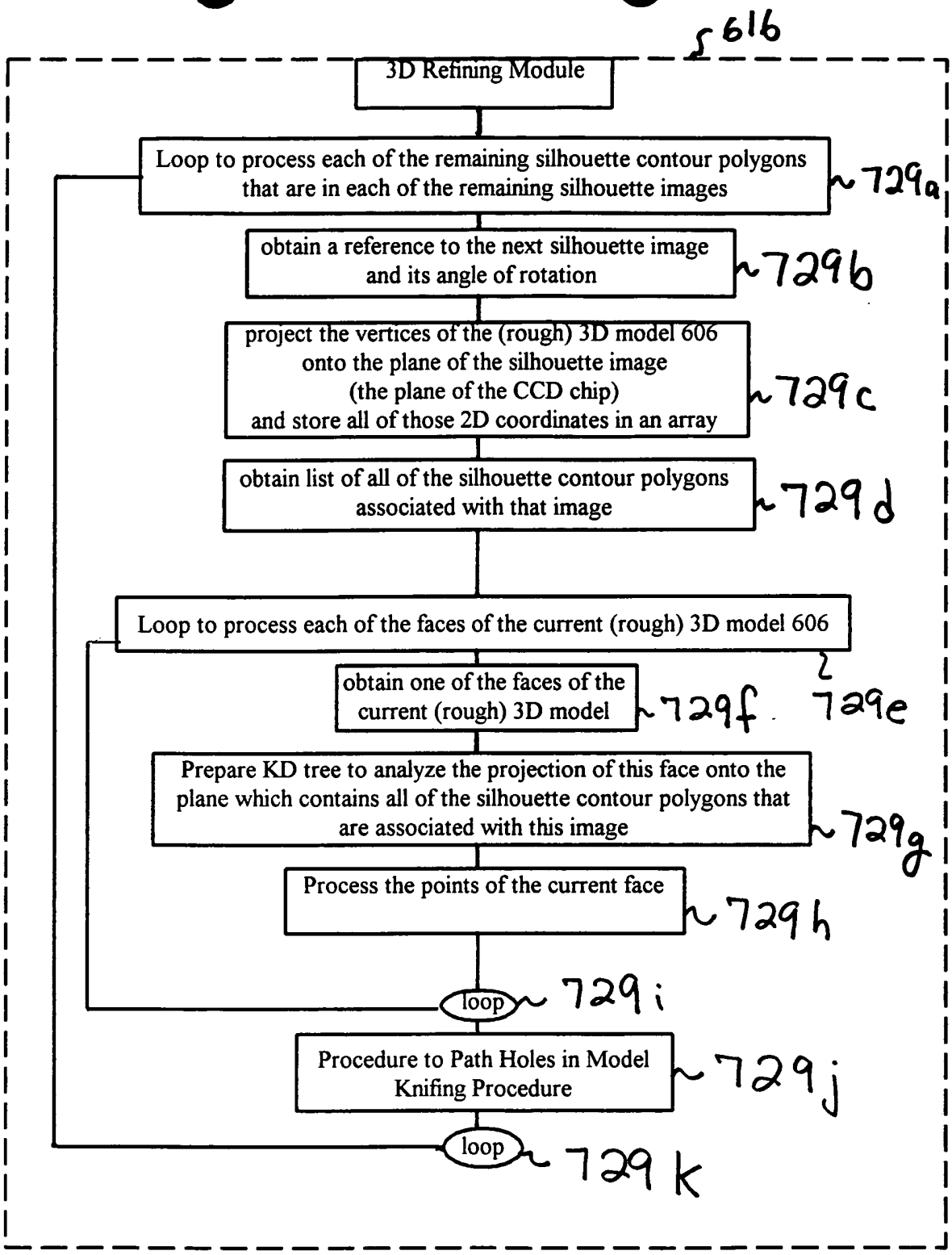

616
3D Refining Module
Loop to process each of the remaining silhouette contour polygons that are in each of the remaining silhouette images
729a
obtain a reference to the next silhouette image and its angle of rotation
729b
project the vertices of the (rough) 3D model 606 onto the plane of the silhouette image (the plane of the CCD chip) and store all of those 2D coordinates in an array
729c
obtain list of all of the silhouette contour polygons associated with that image
729d
Loop to process each of the faces of the current (rough) 3D model 606
729e
obtain one of the faces of the current (rough) 3D model
729f
Prepare KD tree to analyze the projection of this face onto the plane which contains all of the silhouette contour polygons that are associated with this image
729g
Process the points of the current face
729h
loop
729i
Procedure to Path Holes in Model Knifing Procedure
729j
loop
729k

Fig. 29a

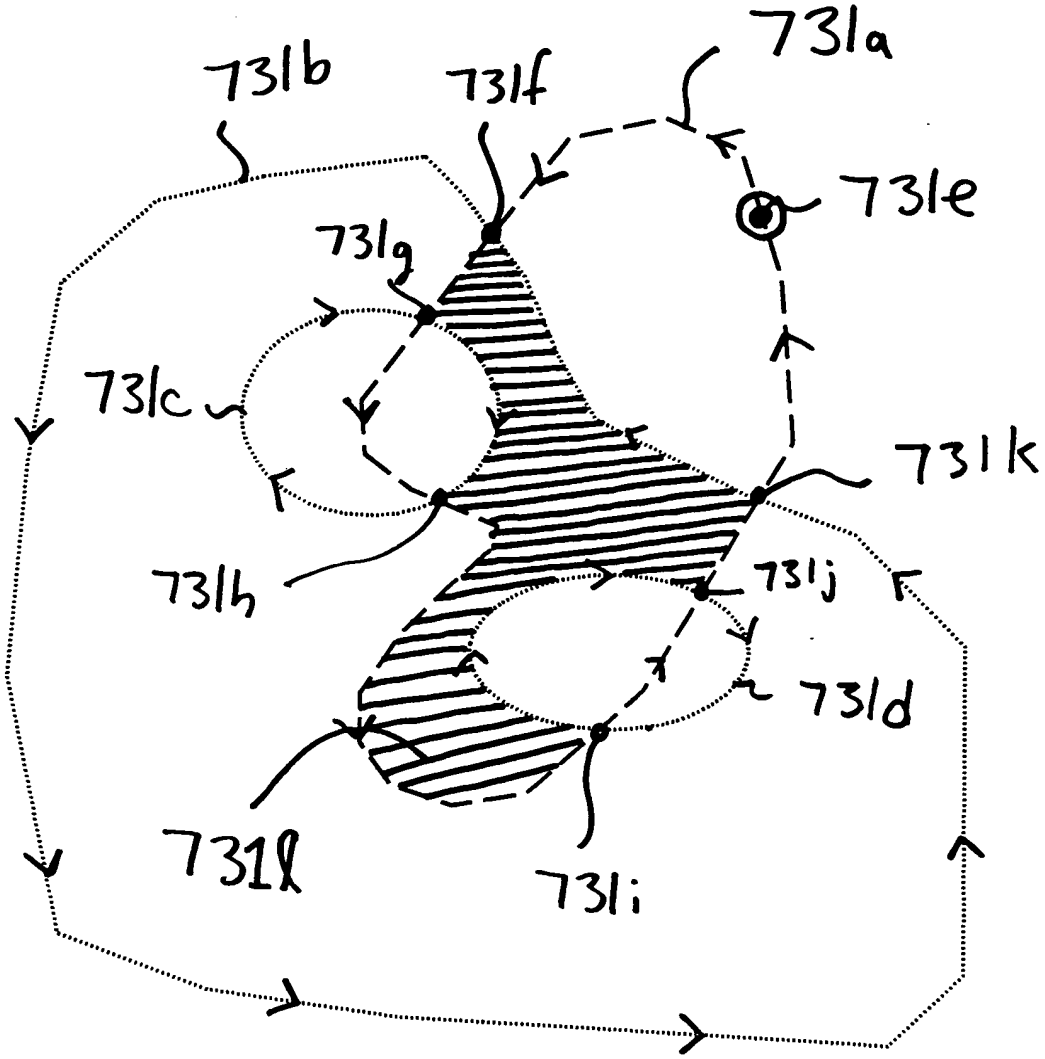


Fig. 29b

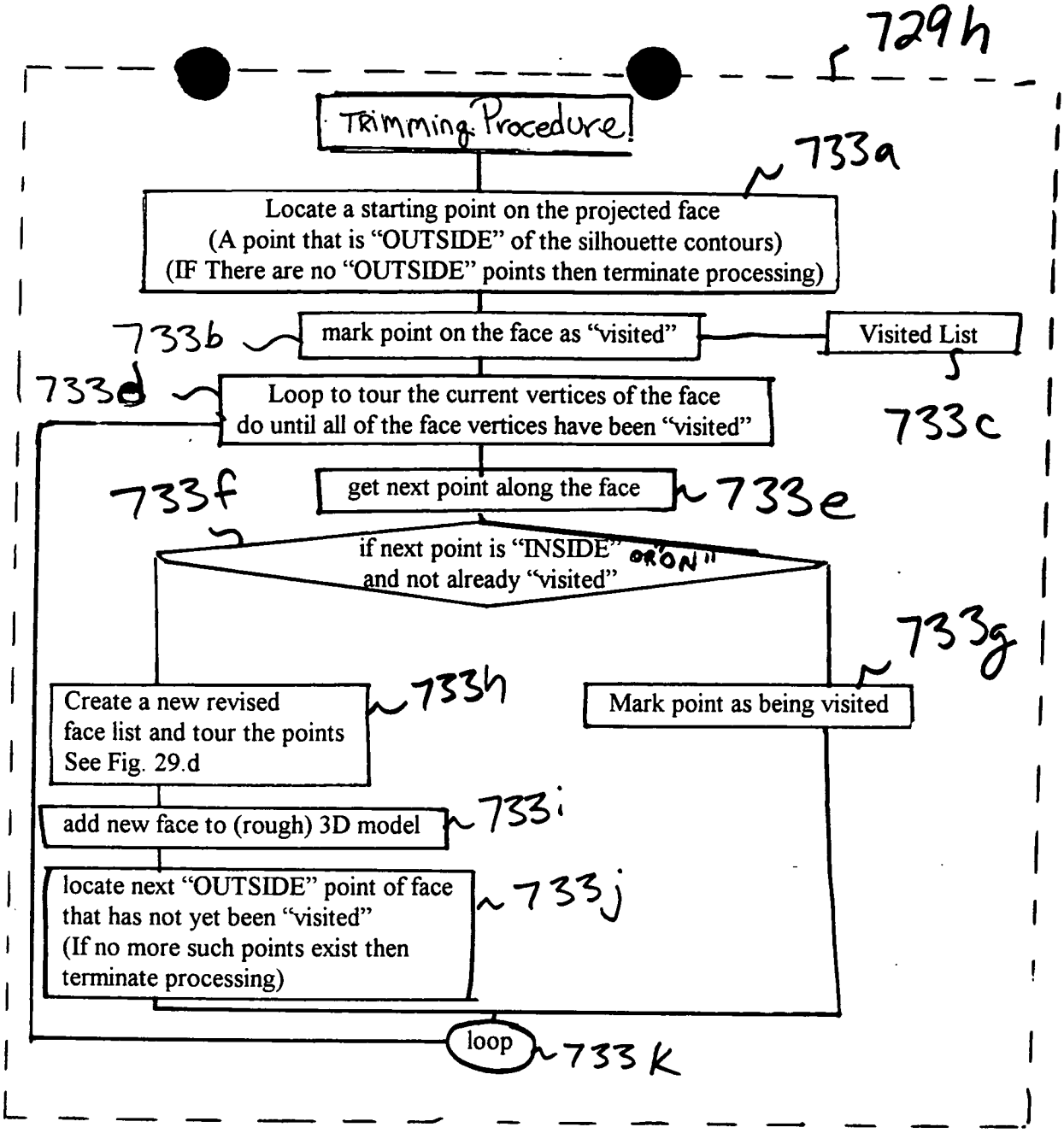

729h
Trimming Procedure
733a
Locate a starting point on the projected face
(A point that is "OUTSIDE" of the silhouette contours)
(IF There are no "OUTSIDE" points then terminate processing)
733b
mark point on the face as "visited"
Visited List
733c
733d
Loop to tour the current vertices of the face
do until all of the face vertices have been "visited"
733f
get next point along the face
733e
if next point is "INSIDE" OR "ON"
and not already "visited"
733g
Mark point as being visited
Create a new revised
face list and tour the points
See Fig. 29.d
733h
add new face to (rough) 3D model
733i
locate next "OUTSIDE" point of face
that has not yet been "visited"
(If no more such points exist then
terminate processing)
733j
loop
733k

Fig. 29c

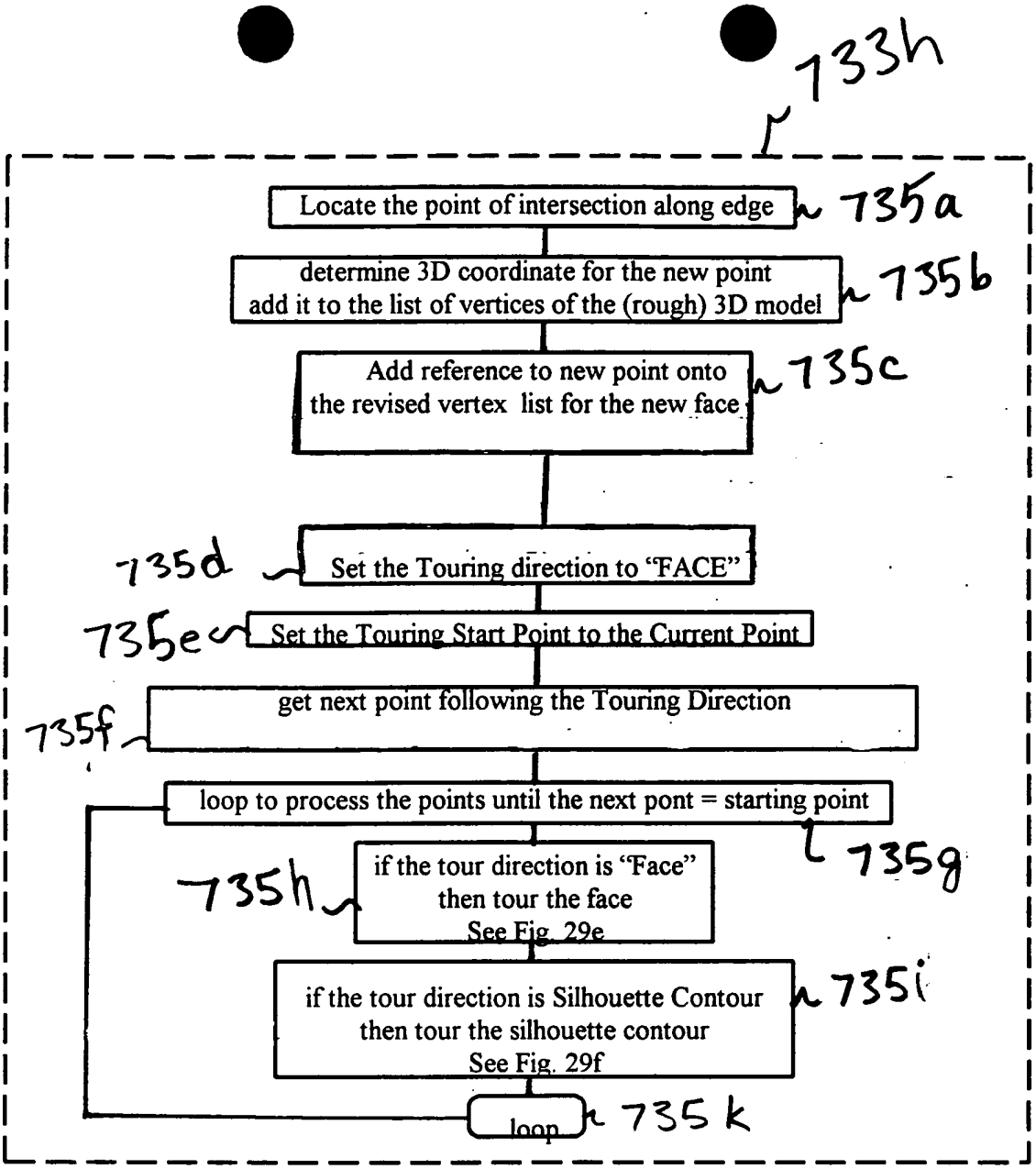

733h
Locate the point of intersection along edge
735a
determine 3D coordinate for the new point
add it to the list of vertices of the (rough) 3D model
735b
Add reference to new point onto
the revised vertex list for the new face
735c
735d
Set the Touring direction to "FACE"
735e
Set the Touring Start Point to the Current Point
735f
get next point following the Touring Direction
loop to process the points until the next pont = starting point
735g
735h
if the tour direction is "Face"
then tour the face
See Fig. 29e
if the tour direction is Silhouette Contour
then tour the silhouette contour
See Fig. 29f
735i
loop
735k

Fig. 29d

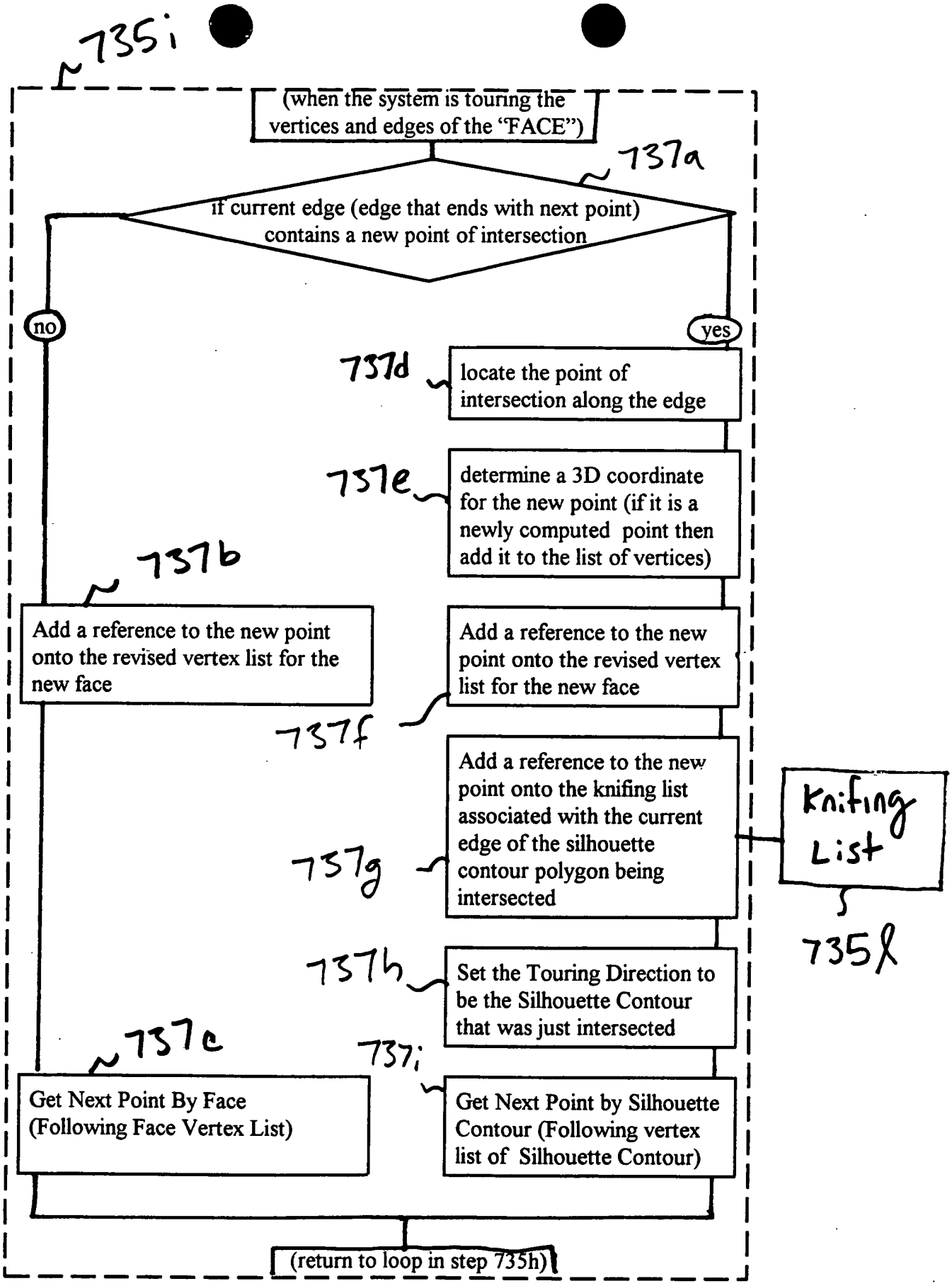

735i
(when the system is touring the vertices and edges of the "FACE")
737a
if current edge (edge that ends with next point) contains a new point of intersection
no
yes
737d
locate the point of intersection along the edge
737e
determine a 3D coordinate for the new point (if it is a newly computed point then add it to the list of vertices)
737b
Add a reference to the new point onto the revised vertex list for the new face
Add a reference to the new point onto the revised vertex list for the new face
737f
Add a reference to the new point onto the knifing list associated with the current edge of the silhouette contour polygon being intersected
737g
Knifing List
735l
737h
Set the Touring Direction to be the Silhouette Contour that was just intersected
737c
737i
Get Next Point By Face (Following Face Vertex List)
Get Next Point by Silhouette Contour (Following vertex list of Silhouette Contour)
(return to loop in step 735h)

Fig. 29e

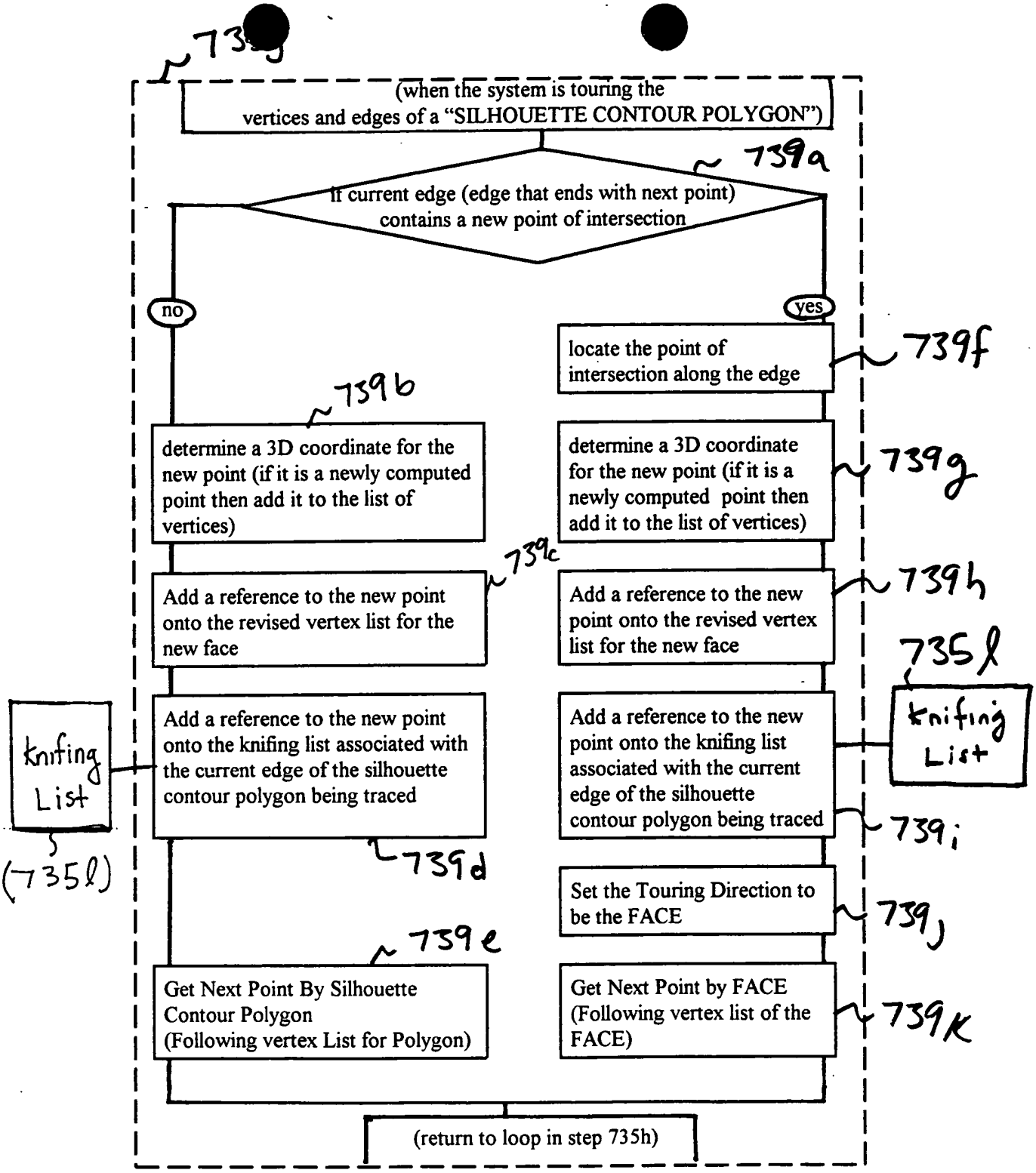
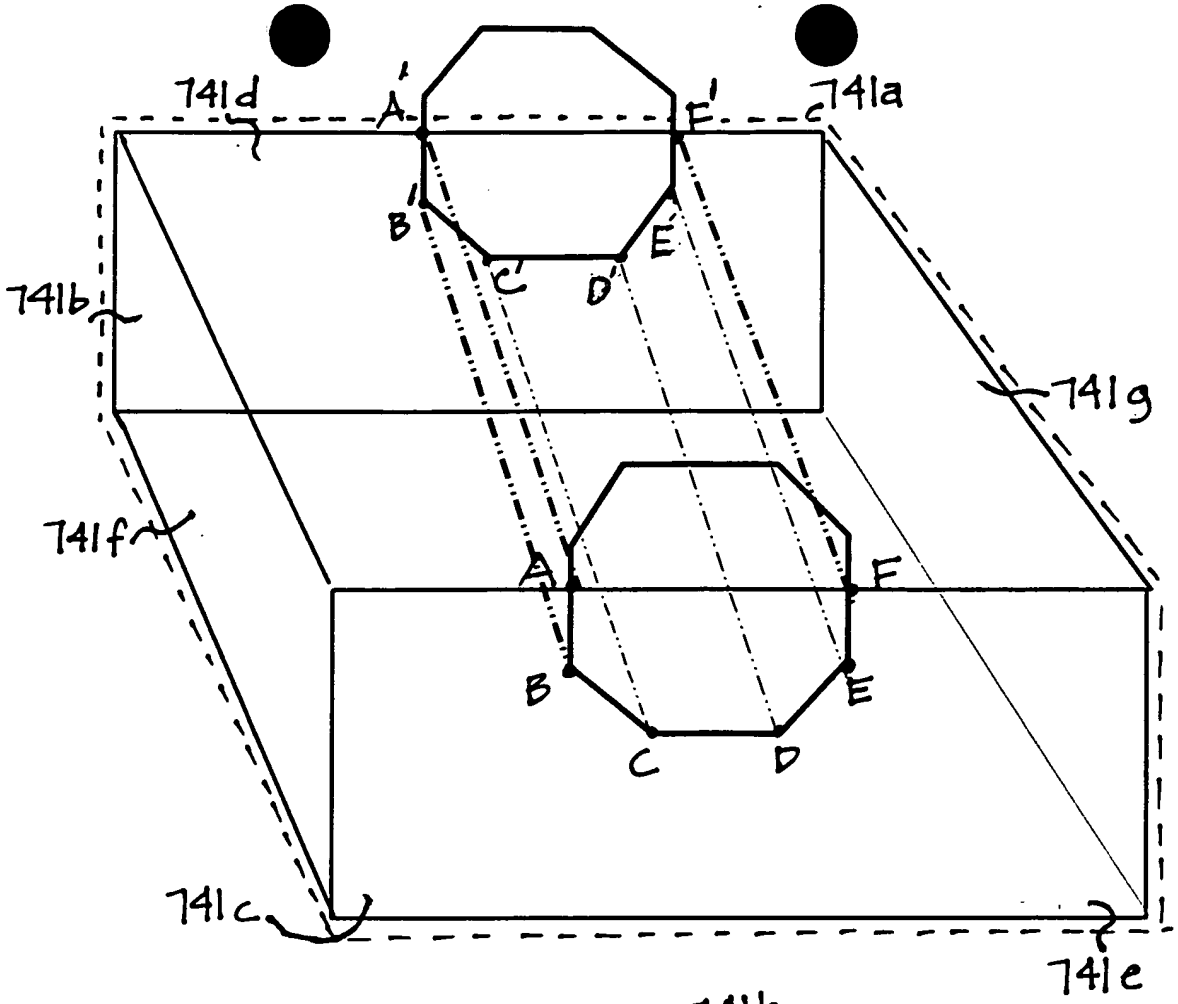
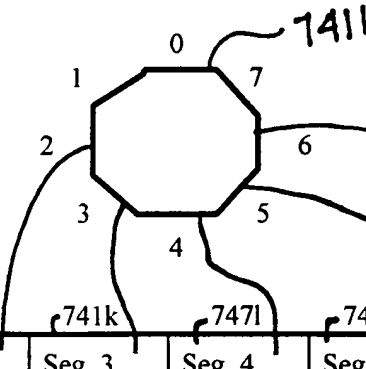

739j
(when the system is touring the vertices and edges of a "SILHOUETTE CONTOUR POLYGON")
739a
If current edge (edge that ends with next point) contains a new point of intersection
no
yes
739b
determine a 3D coordinate for the new point (if it is a newly computed point then add it to the list of vertices)
739c
Add a reference to the new point onto the revised vertex list for the new face
Add a reference to the new point onto the knifing list associated with the current edge of the silhouette contour polygon being traced
739d
739e
Get Next Point By Silhouette Contour Polygon (Following vertex List for Polygon)
Knifing List
(735l)
locate the point of intersection along the edge
739f
determine a 3D coordinate for the new point (if it is a newly computed point then add it to the list of vertices)
739g
Add a reference to the new point onto the revised vertex list for the new face
739h
735l
Knifing List
Add a reference to the new point onto the knifing list associated with the current edge of the silhouette contour polygon being traced
739i
Set the Touring Direction to be the FACE
739j
Get Next Point by FACE (Following vertex list of the FACE)
739k
(return to loop in step 735h)

Fig. 29f

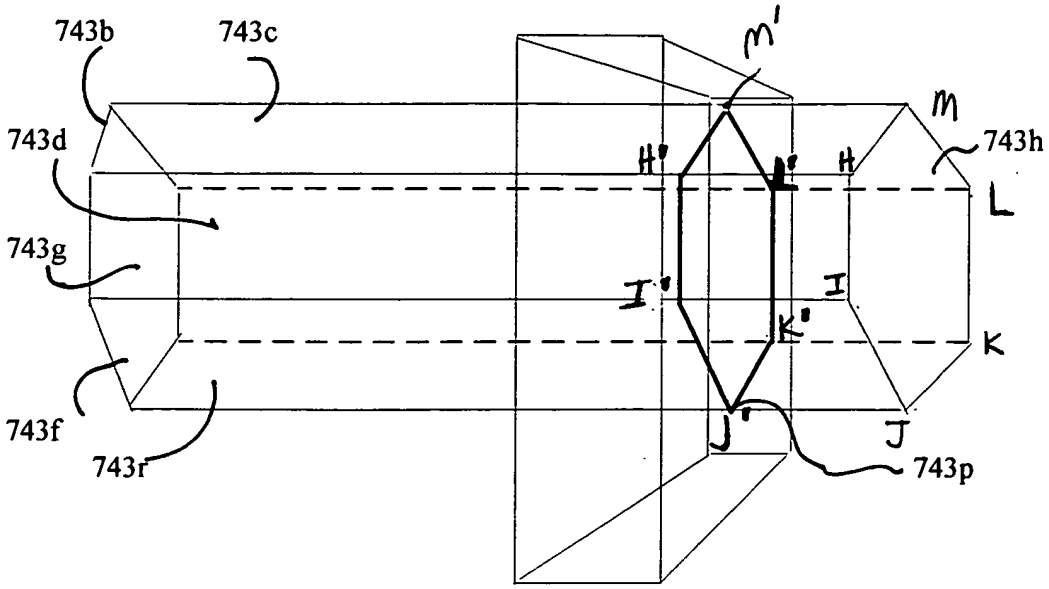




741h	741i	741j	741k	747l	747m	747n	747n
Seg. 0	Seg. 1	Seg. 2	Seg. 3	Seg. 4	Seg. 5	Seg. 6	Seg. 7
		A -B	B-C	C-D	D-E	E-F	
		B'-A'	C'-B'	D'-C'	E'-D'	F'-E'	

741p 741q 741r 741s 741t 741u

Fig. 30a


743b
743c
743d
743g
743f
743r
743h
743p
m'
m
H'
H
L
I'
I
K'
K
J'
J

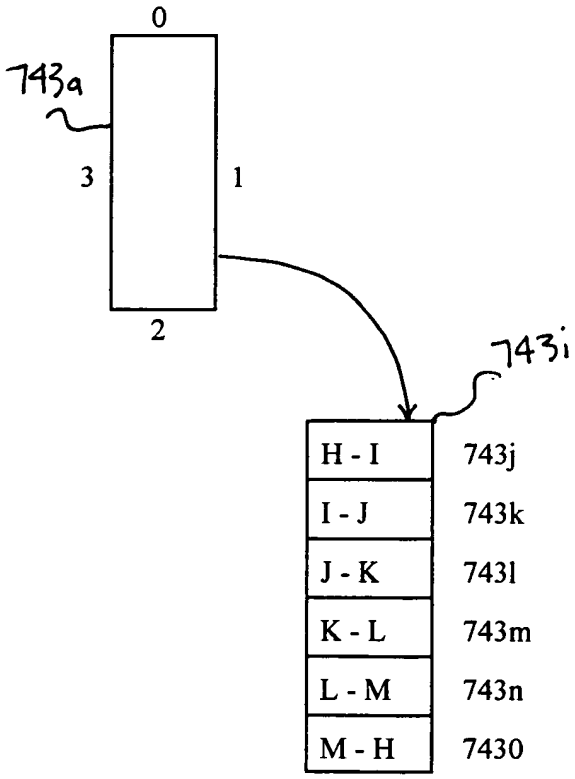

0
743a
3
1
2
743i
H - I
743j
I - J
743k
J - K
743l
K - L
743m
L - M
743n
M - H
7430

Fig. 30b

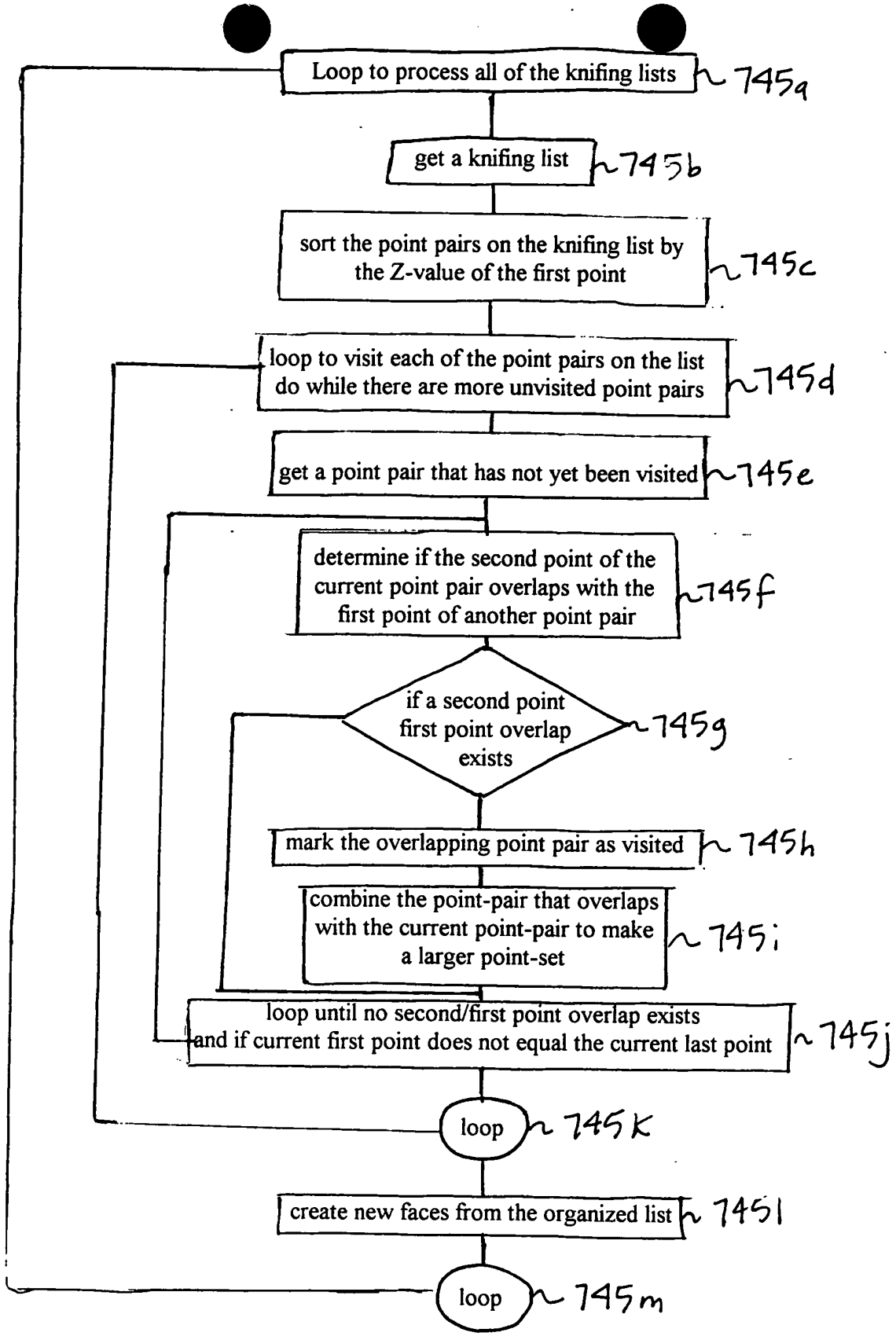

Loop to process all of the knifing lists
745a
get a knifing list
745b
sort the point pairs on the knifing list by the Z-value of the first point
745c
loop to visit each of the point pairs on the list do while there are more unvisited point pairs
745d
get a point pair that has not yet been visited
745e
determine if the second point of the current point pair overlaps with the first point of another point pair
745f
if a second point first point overlap exists
745g
mark the overlapping point pair as visited
745h
combine the point-pair that overlaps with the current point-pair to make a larger point-set
745i
loop until no second/first point overlap exists and if current first point does not equal the current last point
745j
loop
745k
create new faces from the organized list
745l
loop
745m

Fig. 31

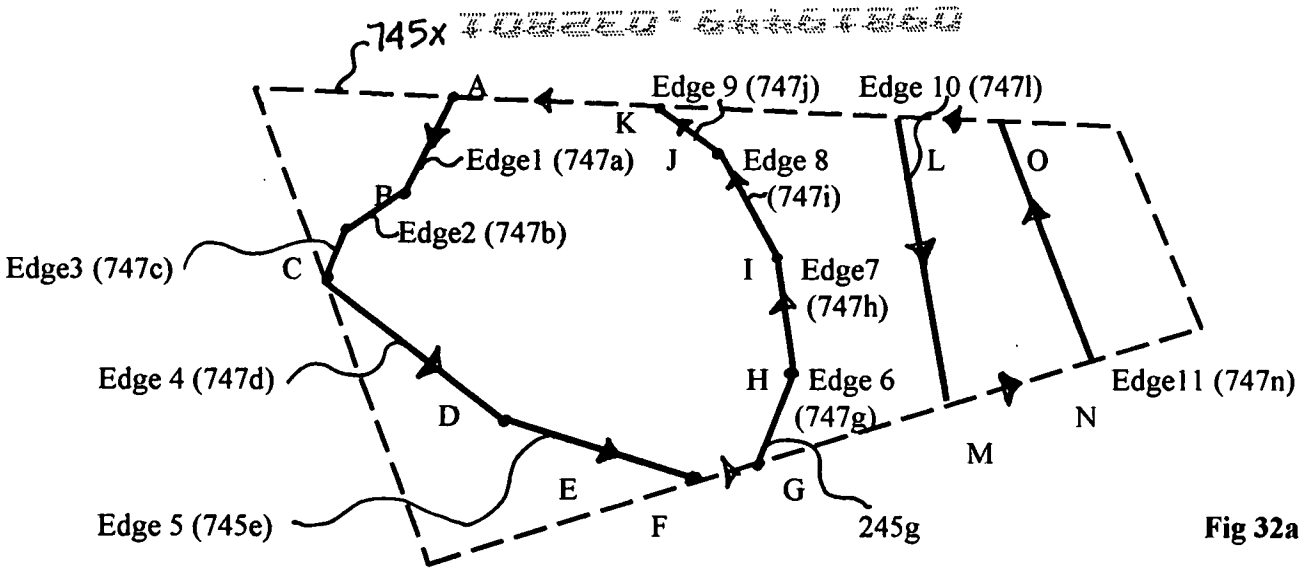


Fig 32a

747a	747e	747i	747b	747g	747d	747h	747c	747j	747l	747n
Edge1	Edge 5	Edge 8	Edge 2	Edge 6	Edge 4	Edge 7	Edge 3	Edge 9	Edge 10	Edge 11
A:30,80,50	E:20,20,43	I:70,60,90	B:20,65,40	G:60,15,80	D:15,42,32	H:70,30,92	C:10,50,30	J:60,80,75	L:80,55,100	N:90,26,115
B:20,65,40	F:25,10,50	J:60,80,75	C:10,50,30	H:70,30,92	E:20,20,43	I:70,60,90	D:15,42,32	K:50,85,70	M:82,25,110	O:95,40,120

Fig 32b

745c	745d	745b	745e	745a	745j	745g	747i	747h	747l	747n
Edge 3	Edge 4	Edge 2	Edge 5	Edge1	Edge 6	Edge 9	Edge 7	Edge 8	Edge 10	Edge 11
C:10,50,30	D:15,42,32	B:20,65,40	E:20,20,43	A:30,80,50	J:60,80,75	G:60,15,80	I:70,60,90	H:70,30,92	L:80,55,100	N:90,26,115
D:15,42,32	E:20,20,43	C:10,50,30	F:25,10,50	B:20,65,40	K:50,85,70	H:70,30,92	J:60,80,75	I:70,60,90	M:82,25,110	O:95,40,120

Fig. 33

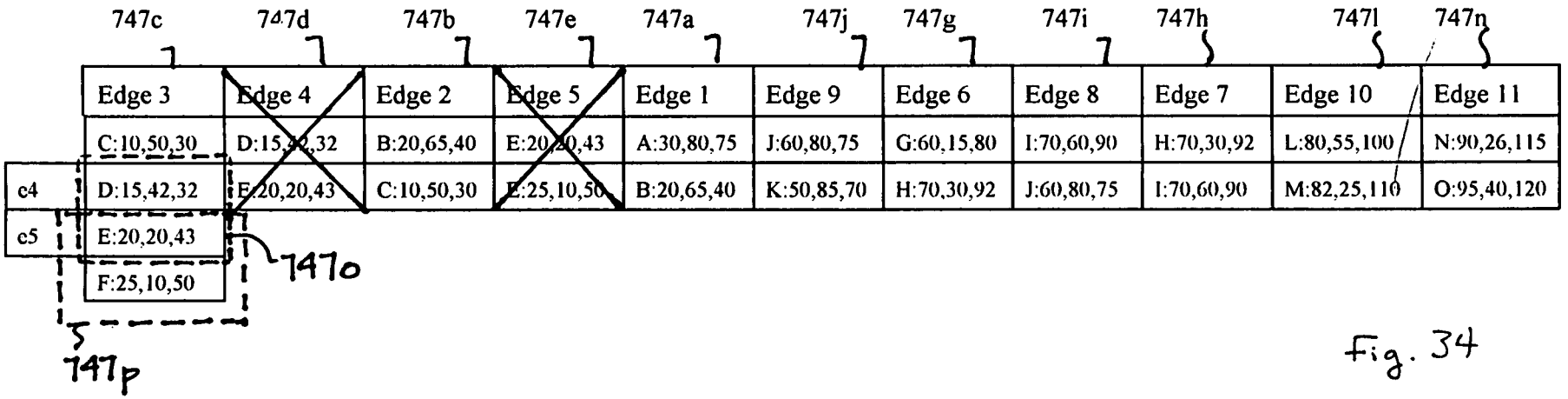


Fig. 34

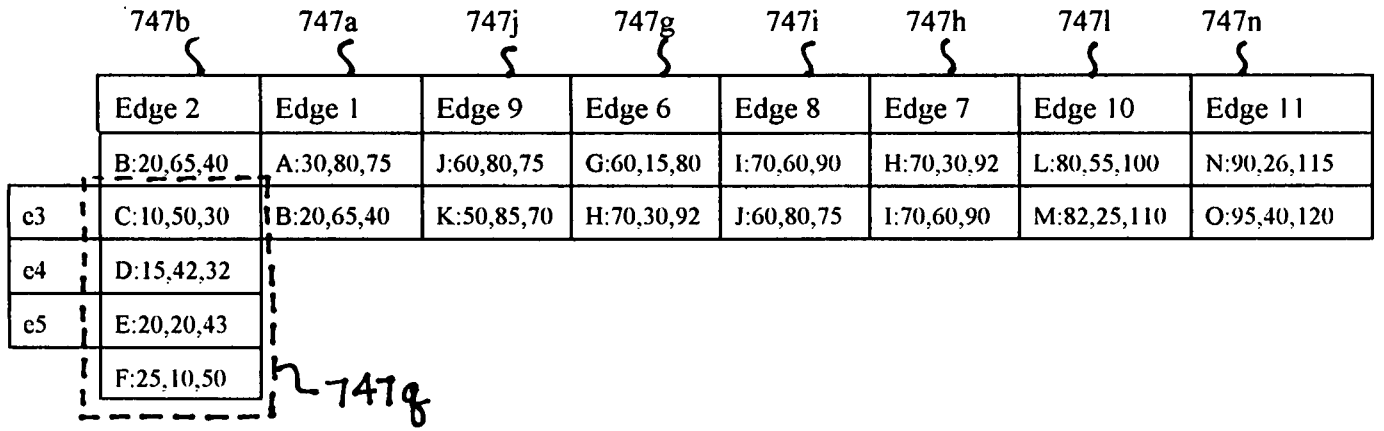


Fig. 35

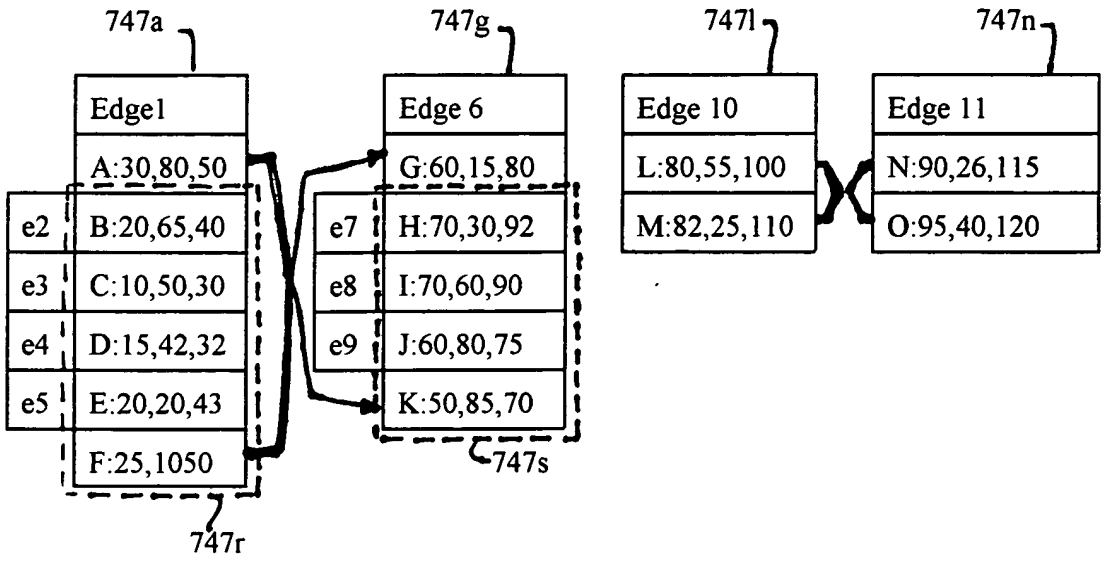

747a
747g
747l
747n
Edge1
A:30,80,50
e2
B:20,65,40
e3
C:10,50,30
e4
D:15,42,32
e5
E:20,20,43
F:25,1050
747r
Edge 6
G:60,15,80
e7
H:70,30,92
e8
I:70,60,90
e9
J:60,80,75
K:50,85,70
747s
Edge 10
L:80,55,100
M:82,25,110
Edge 11
N:90,26,115
O:95,40,120

Fig. 36

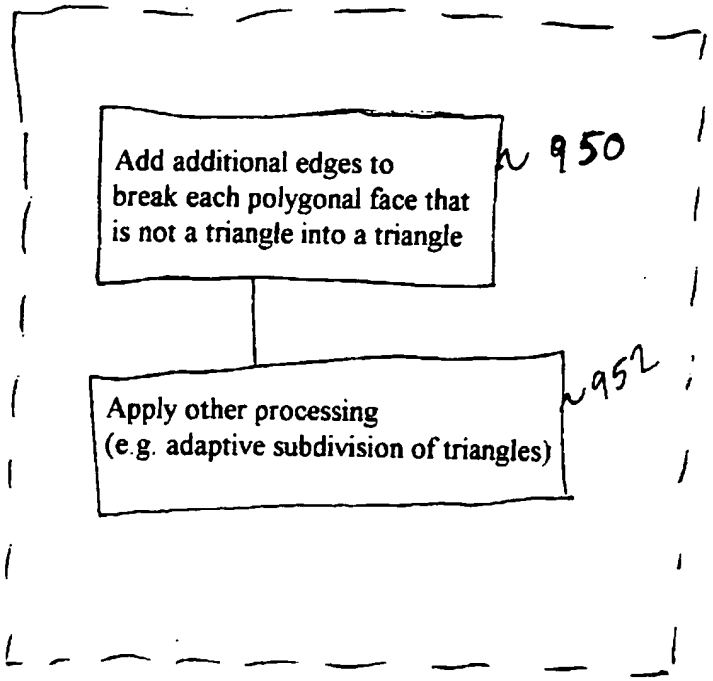

Add additional edges to break each polygonal face that is not a triangle into a triangle
950
Apply other processing (e.g. adaptive subdivision of triangles)
952

Fig. 39

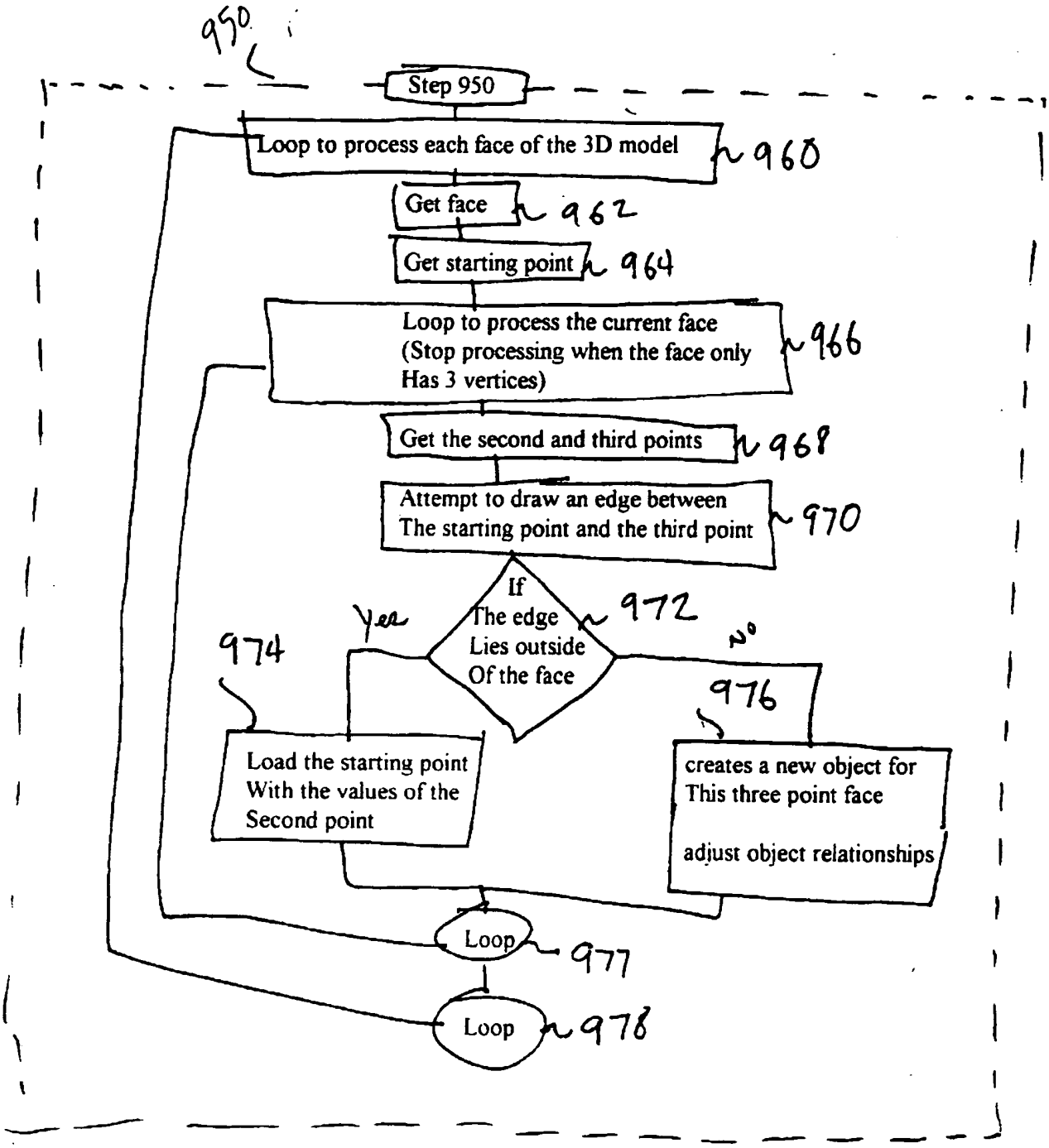

950
Step 950
Loop to process each face of the 3D model
960
Get face
962
Get starting point
964
Loop to process the current face
(Stop processing when the face only
Has 3 vertices)
966
Get the second and third points
968
Attempt to draw an edge between
The starting point and the third point
970
If
The edge
Lies outside
Of the face
972
Yes
No
974
976
Load the starting point
With the values of the
Second point
creates a new object for
This three point face
adjust object relationships
Loop
977
Loop
978

Fig. 38

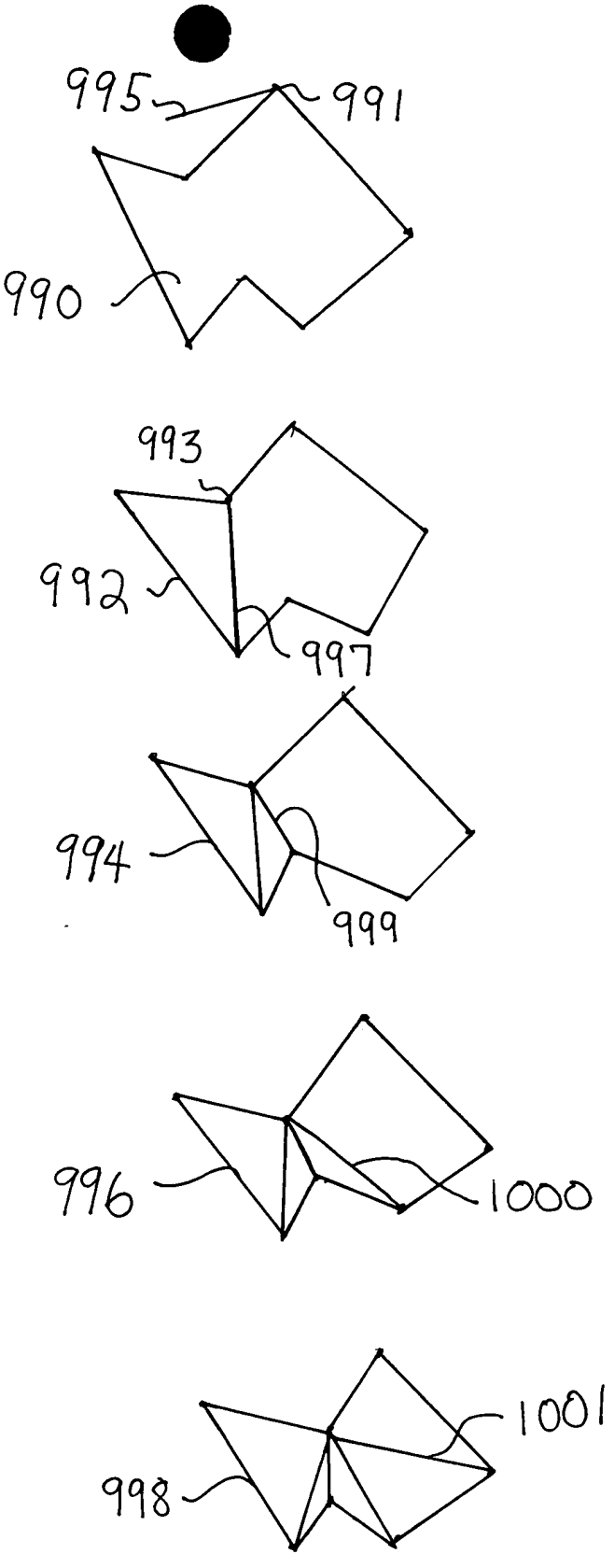


Fig. 39

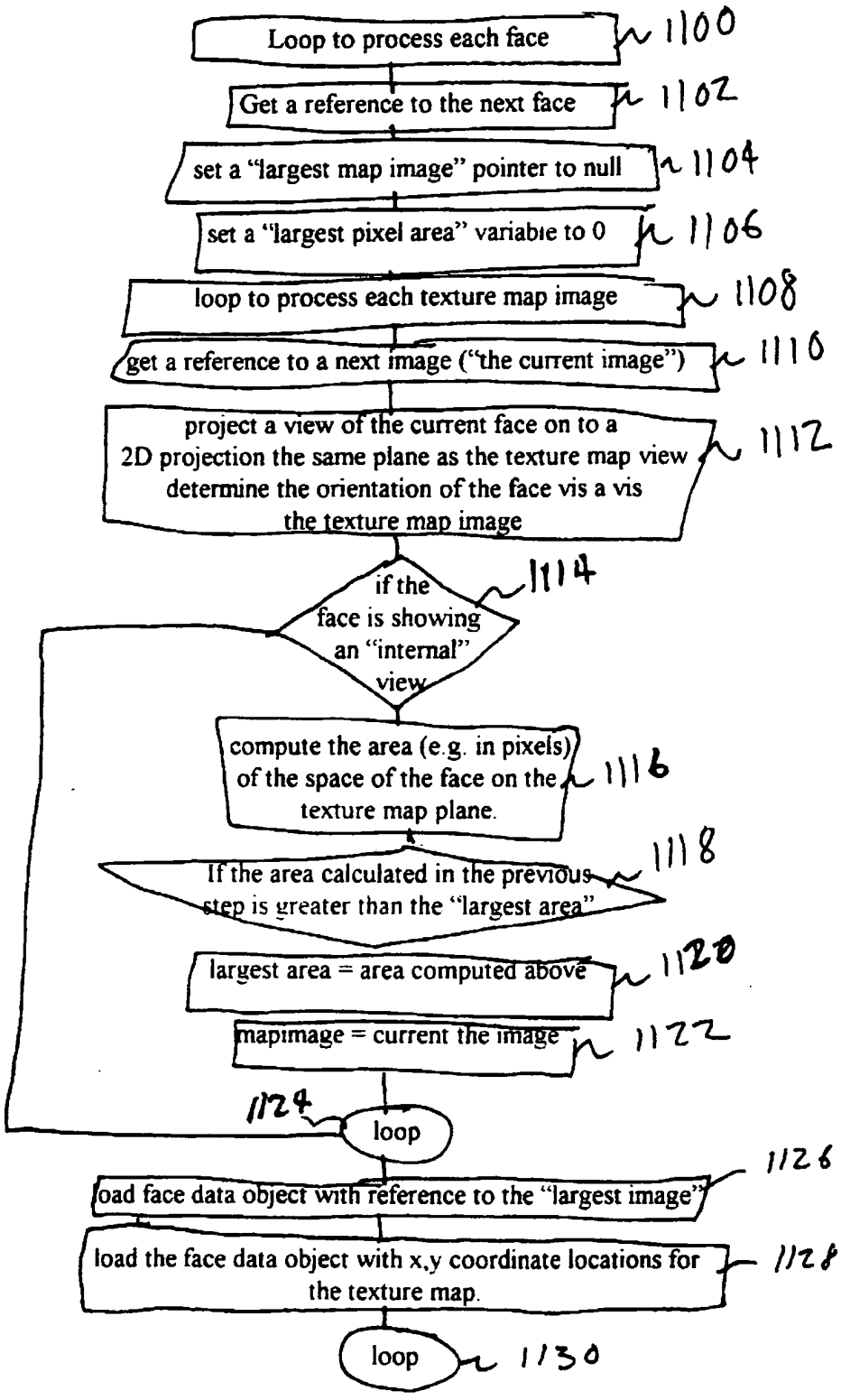

Loop to process each face
1100
Get a reference to the next face
1102
set a "largest map image" pointer to null
1104
set a "largest pixel area" variable to 0
1106
114
loop to process each texture map image
1108
get a reference to a next image ("the current image")
1110
project a view of the current face on to a 2D projection the same plane as the texture map view determine the orientation of the face vis a vis the texture map image
1112
if the face is showing an "internal" view
1114
compute the area (e.g. in pixels) of the space of the face on the texture map plane.
1116
If the area calculated in the previous step is greater than the "largest area"
1118
largest area = area computed above
1120
mapimage = current the image
1122
1124
loop
load face data object with reference to the "largest image"
1126
load the face data object with x,y coordinate locations for the texture map.
1128
loop
1130

Fig. 40

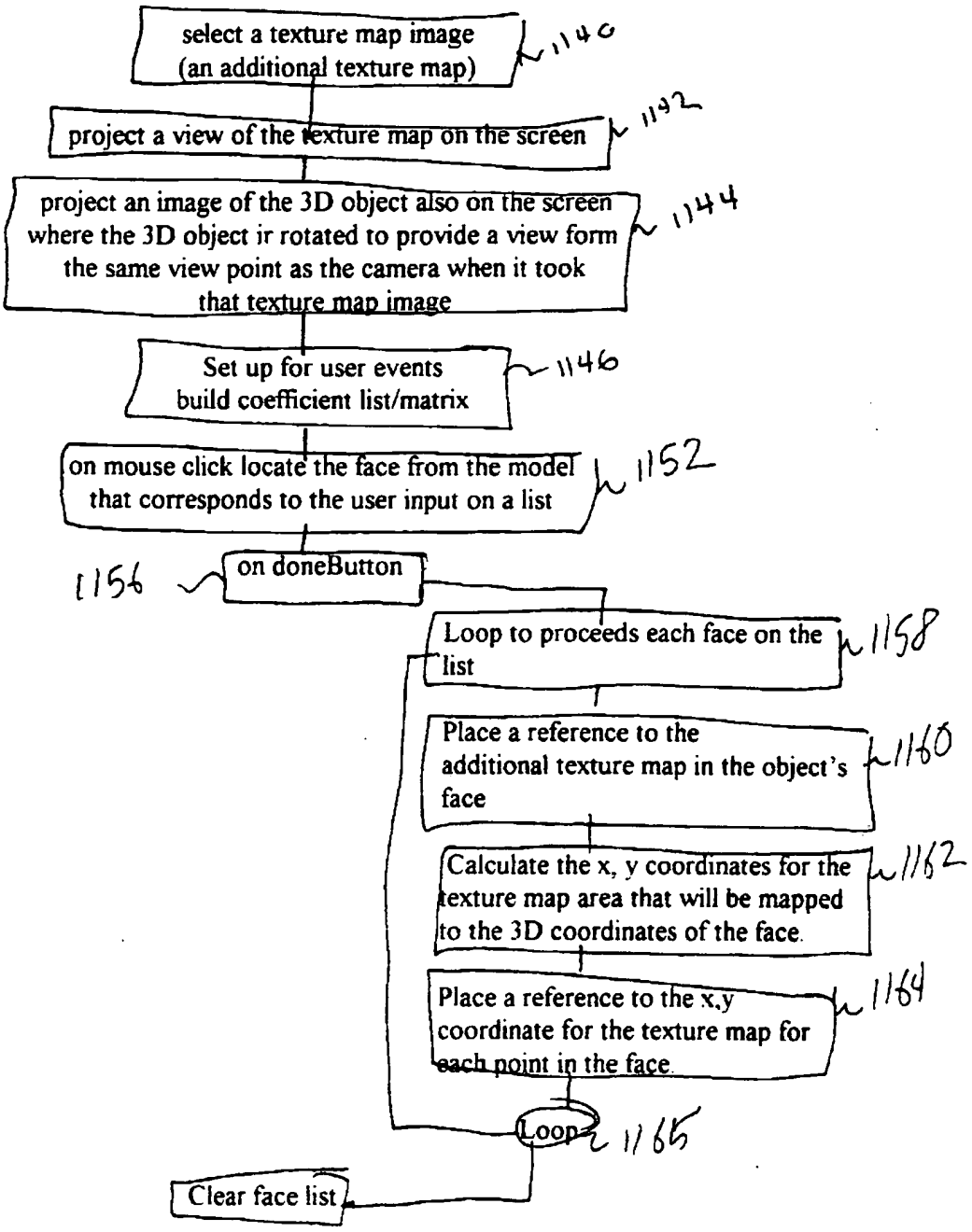

select a texture map image (an additional texture map)
1140
project a view of the texture map on the screen
1142
project an image of the 3D object also on the screen where the 3D object ir rotated to provide a view form the same view point as the camera when it took that texture map image
1144
Set up for user events build coefficient list/matrix
1146
on mouse click locate the face from the model that corresponds to the user input on a list
1152
1156
on doneButton
Loop to proceeds each face on the list
1158
Place a reference to the additional texture map in the object's face
1160
Calculate the x, y coordinates for the texture map area that will be mapped to the 3D coordinates of the face.
1162
Place a reference to the x,y coordinate for the texture map for each point in the face.
1164
Loop
1165
Clear face list

Fig. 41

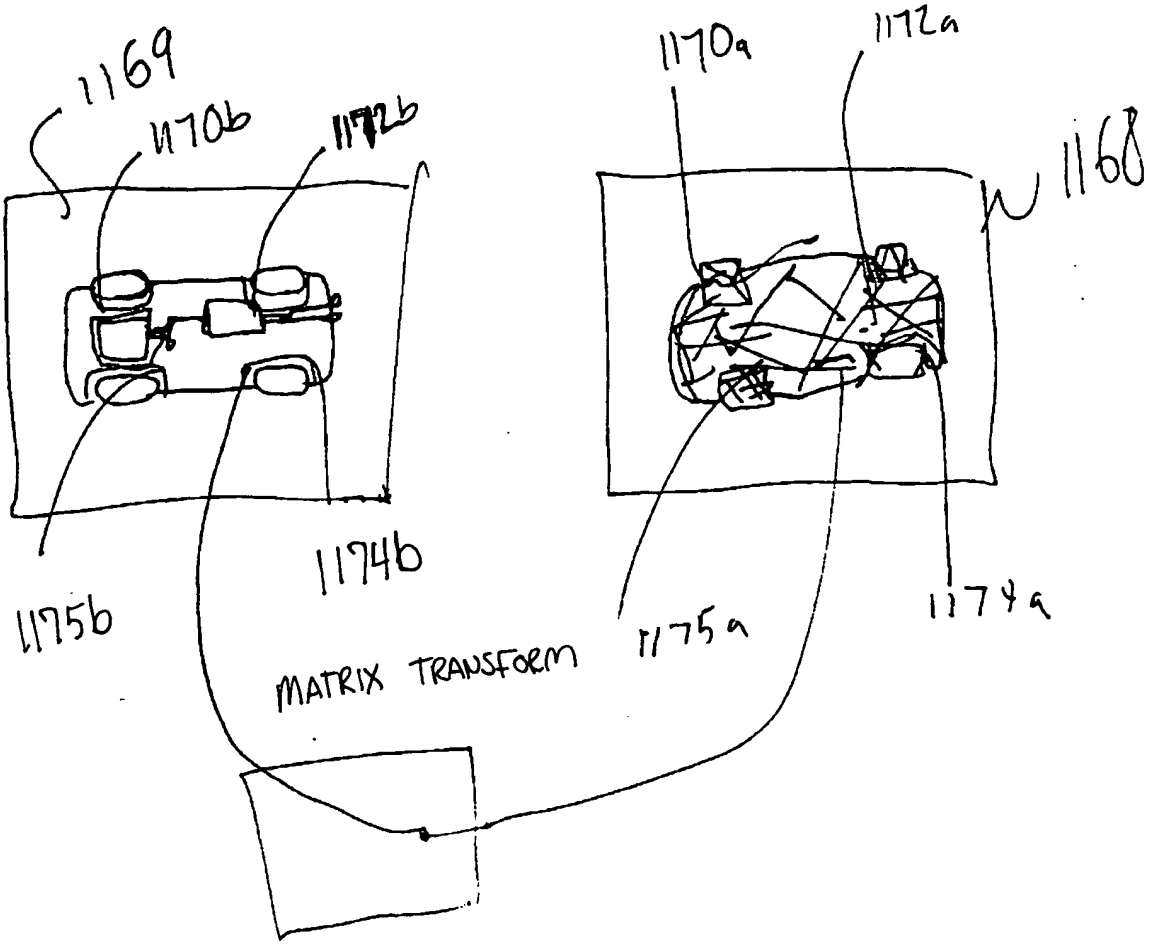

1169
1170b
1172b
1168
1170a
1172a
1175b
1174b
MATRIX TRANSFORM
1175a
1174a

Fig. 42

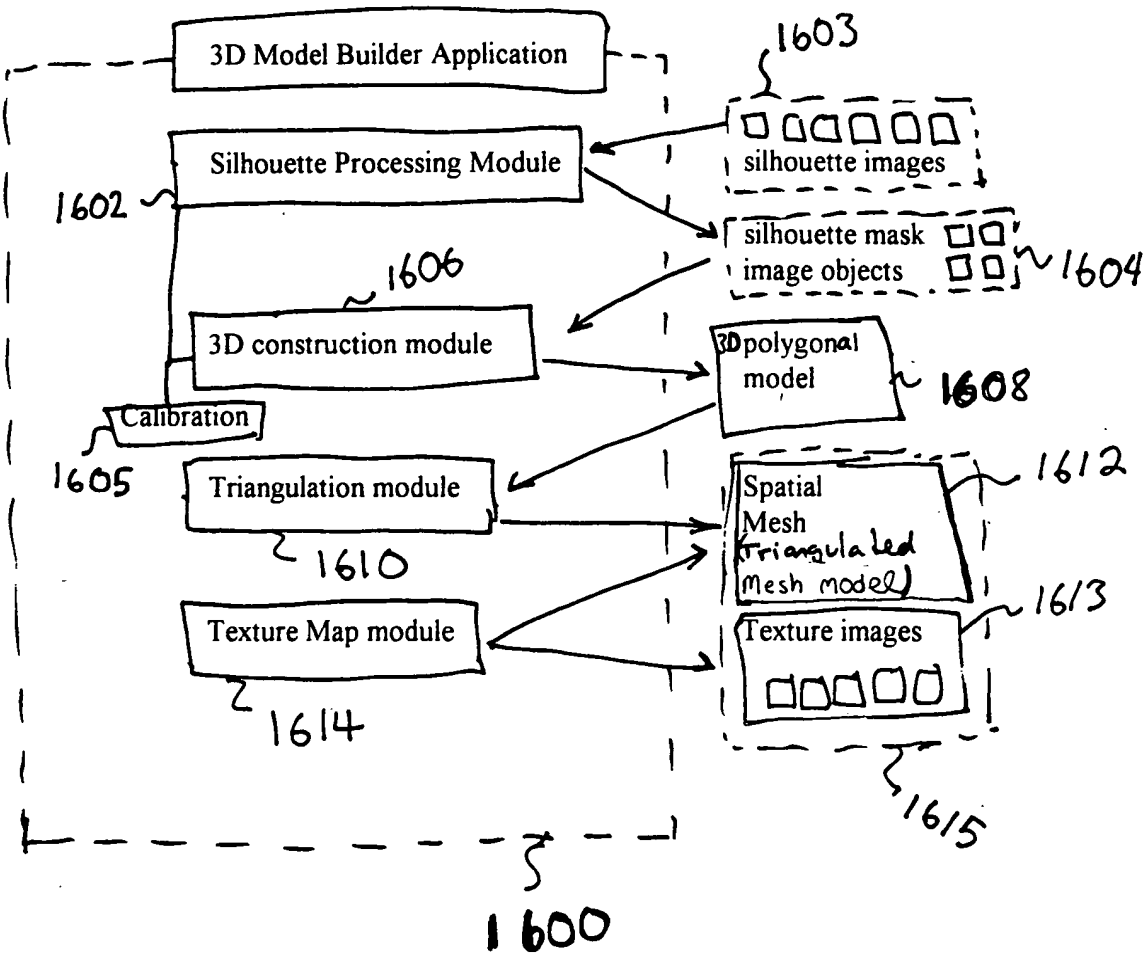

3D Model Builder Application
1603
silhouette images
Silhouette Processing Module
1602
silhouette mask
image objects
1604
1606
3D construction module
3D polygonal model
1608
Calibration
1605
Triangulation module
1610
Spatial Mesh (Triangulated Mesh model)
1612
Texture Map module
1614
Texture images
1613
1615
1600

Fig. 43

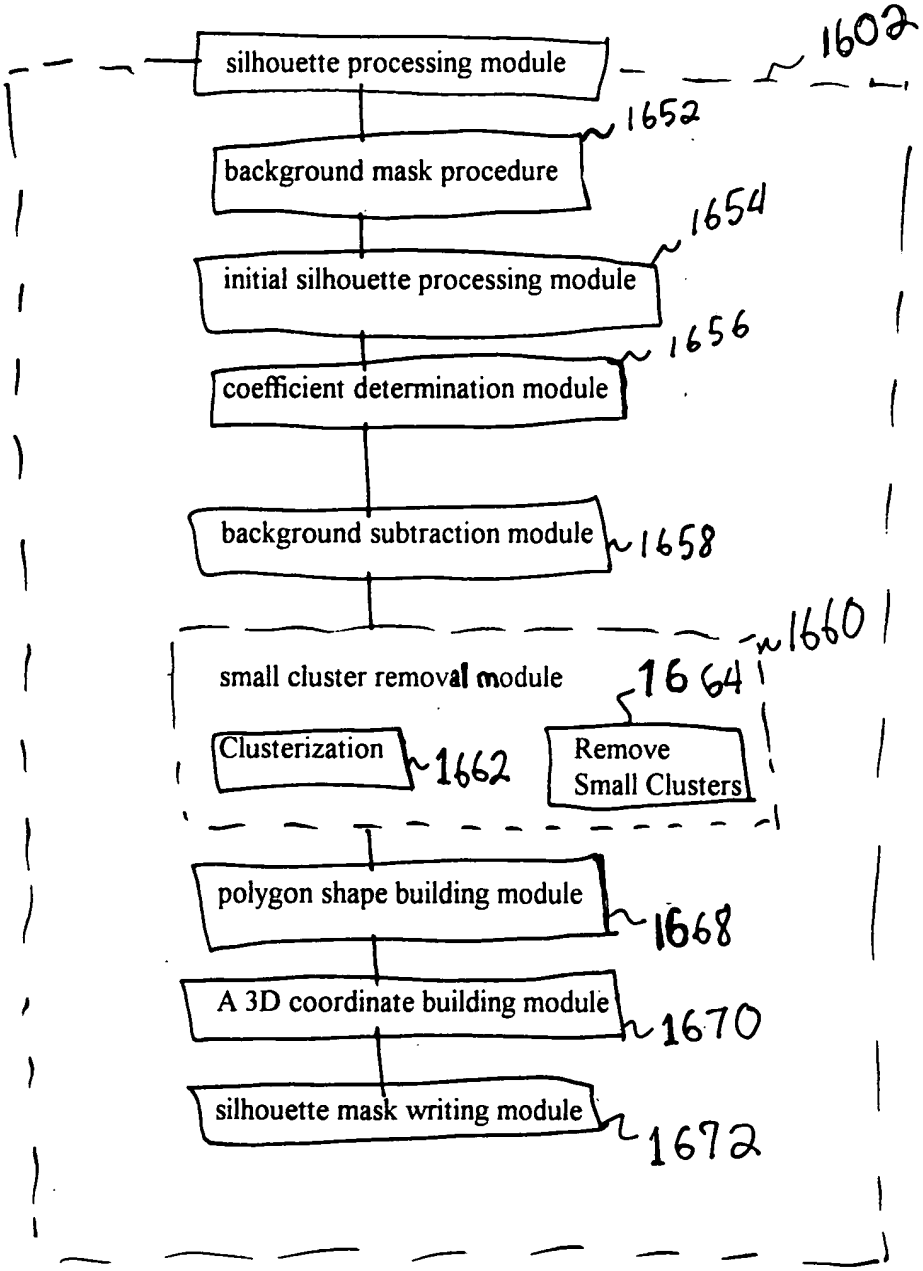

silhouette processing module
1602
background mask procedure
1652
initial silhouette processing module
1654
coefficient determination module
1656
background subtraction module
1658
small cluster removal module
1660
Clusterization
1662
Remove Small Clusters
1664
polygon shape building module
1668
A 3D coordinate building module
1670
silhouette mask writing module
1672

Fig. 44

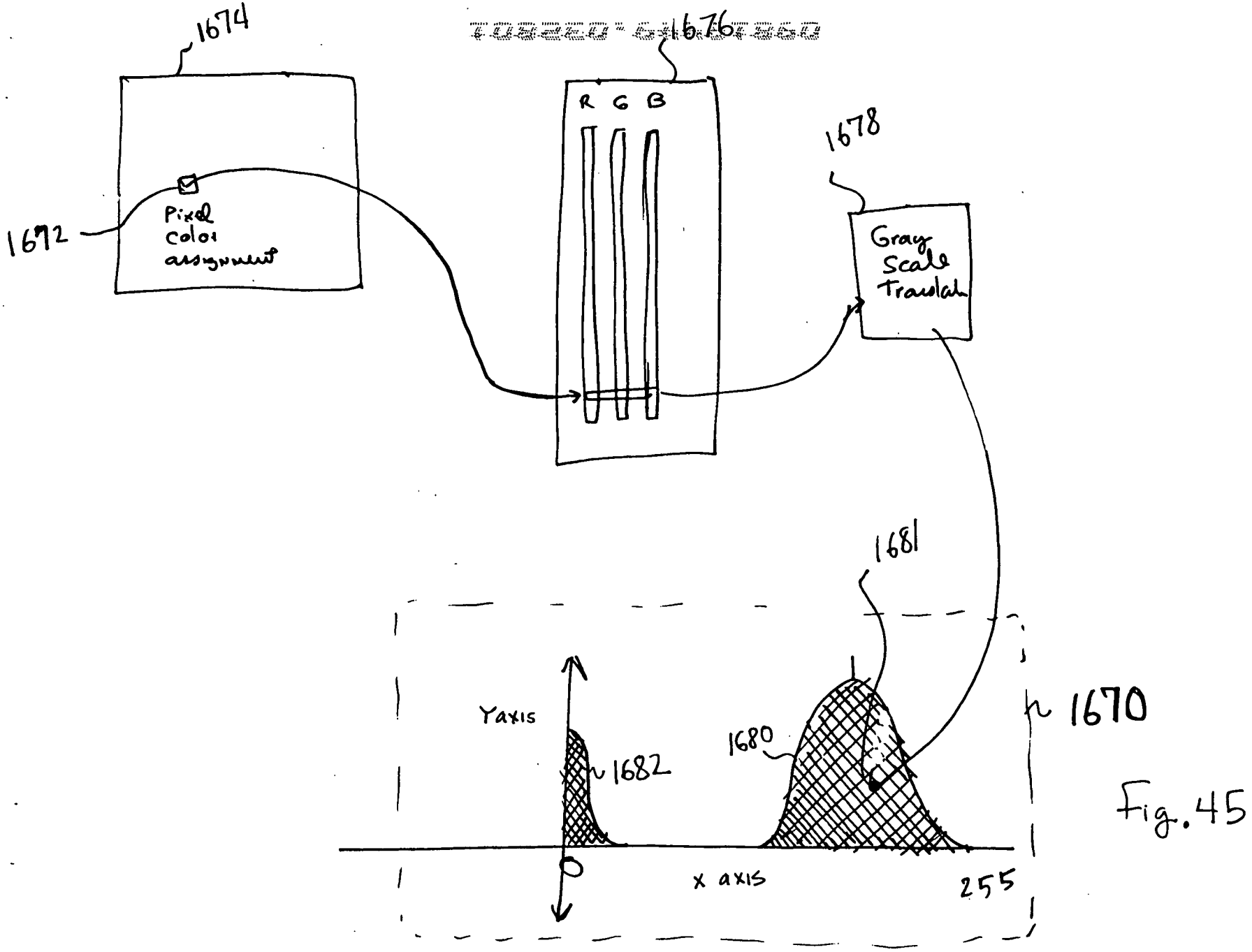

1674
1676
R G B
1678
1672
Pixel
color
assignment
Gray
Scale
Translat
1681
1670
Y axis
1680
1682
0
x axis
255

Fig. 45

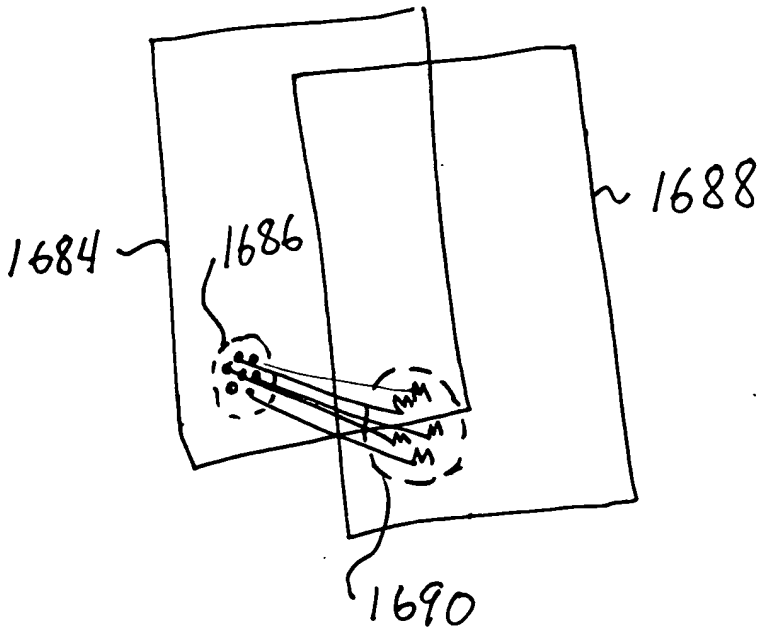


Fig. 46

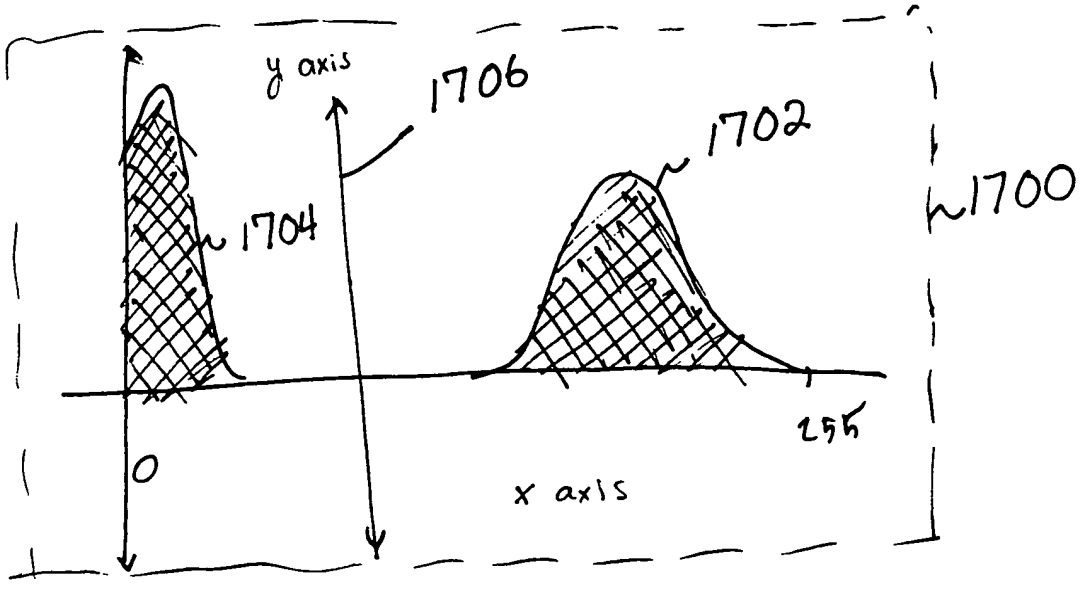


Fig. 47

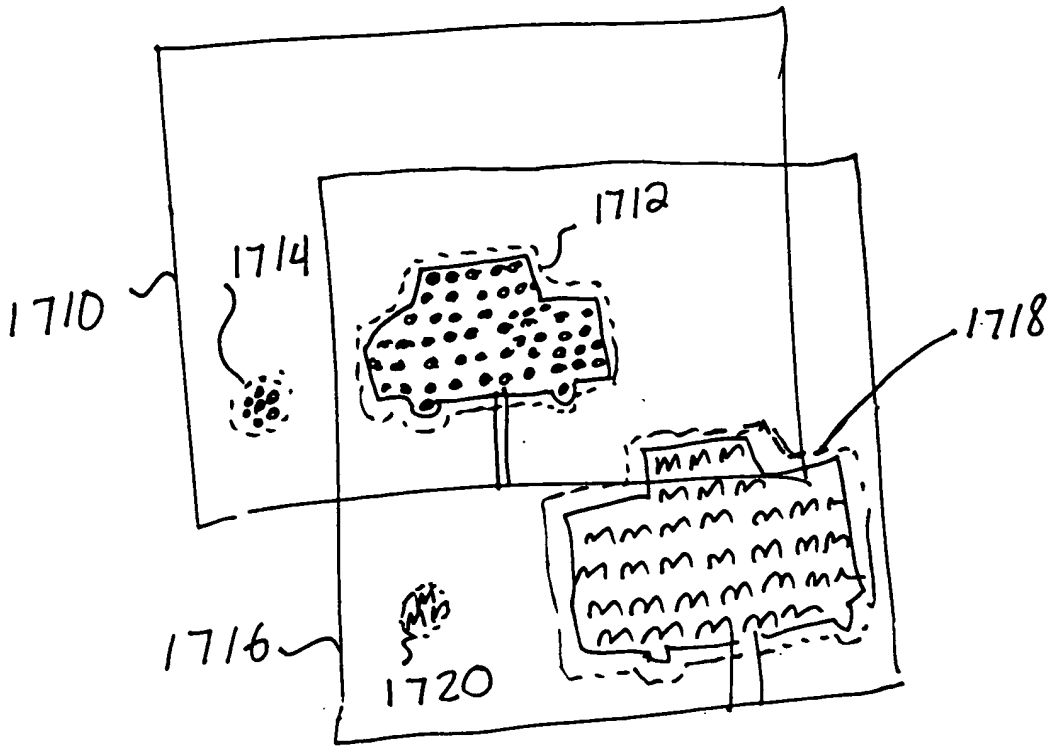

1710
1712
1714
1716
1718
1720

Fig. 48

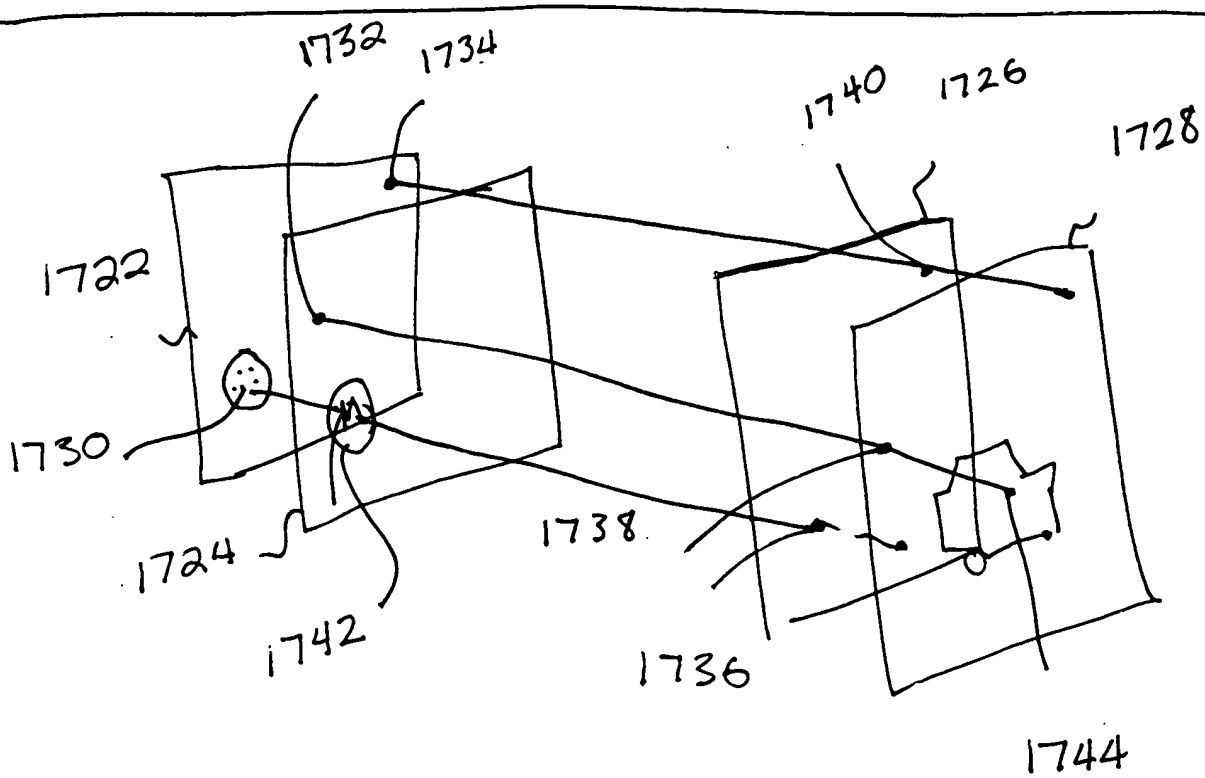

1722
1724
1726
1728
1730
1732
1734
1736
1738
1740
1742
1744

Fig. 49

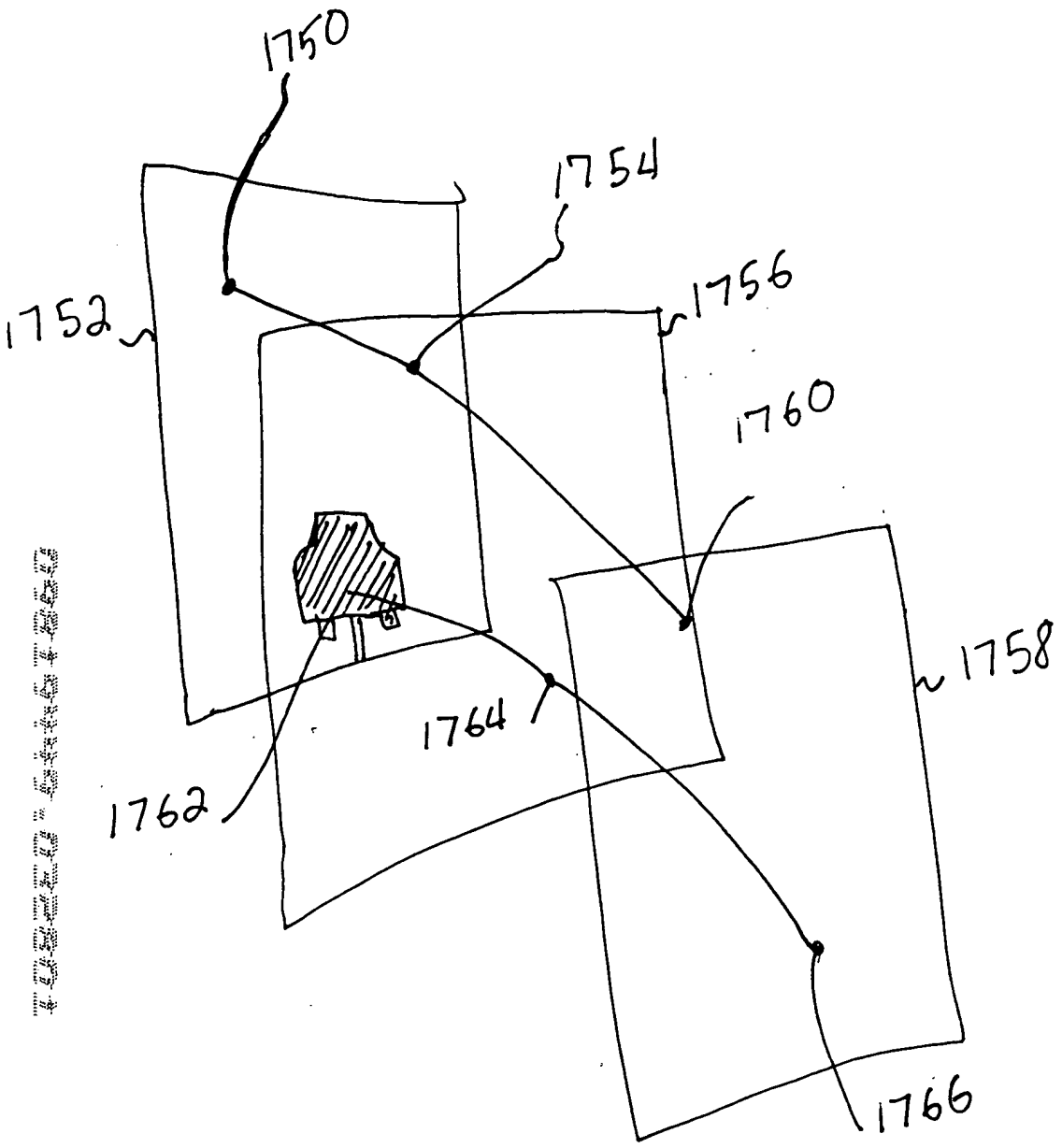


Fig. 50

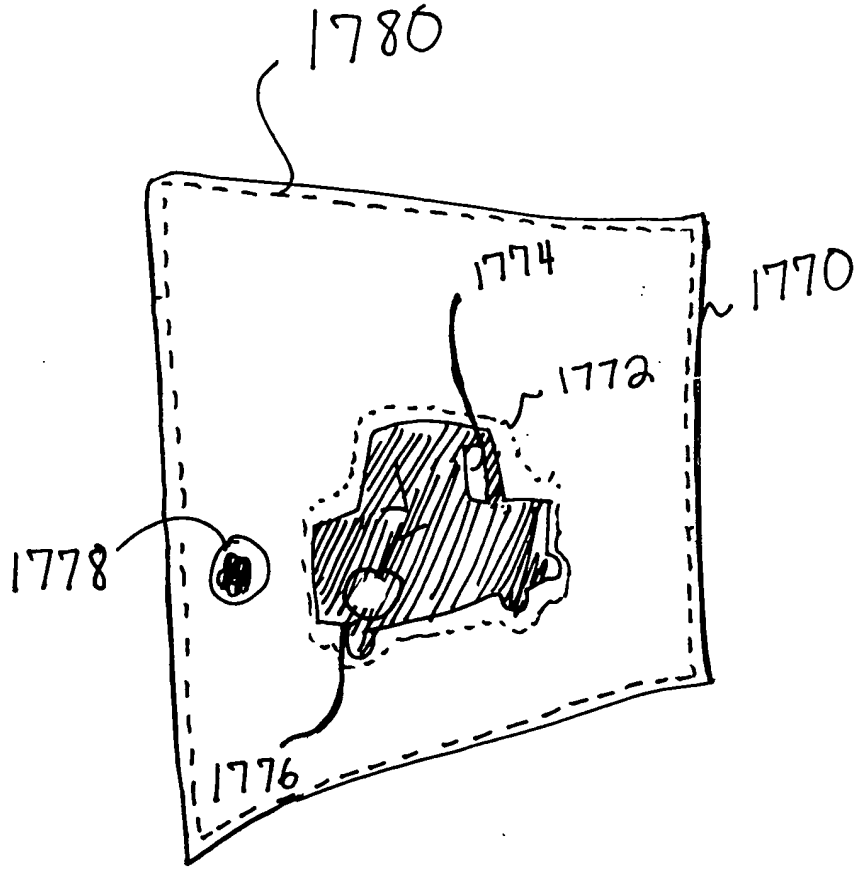


Fig. 51

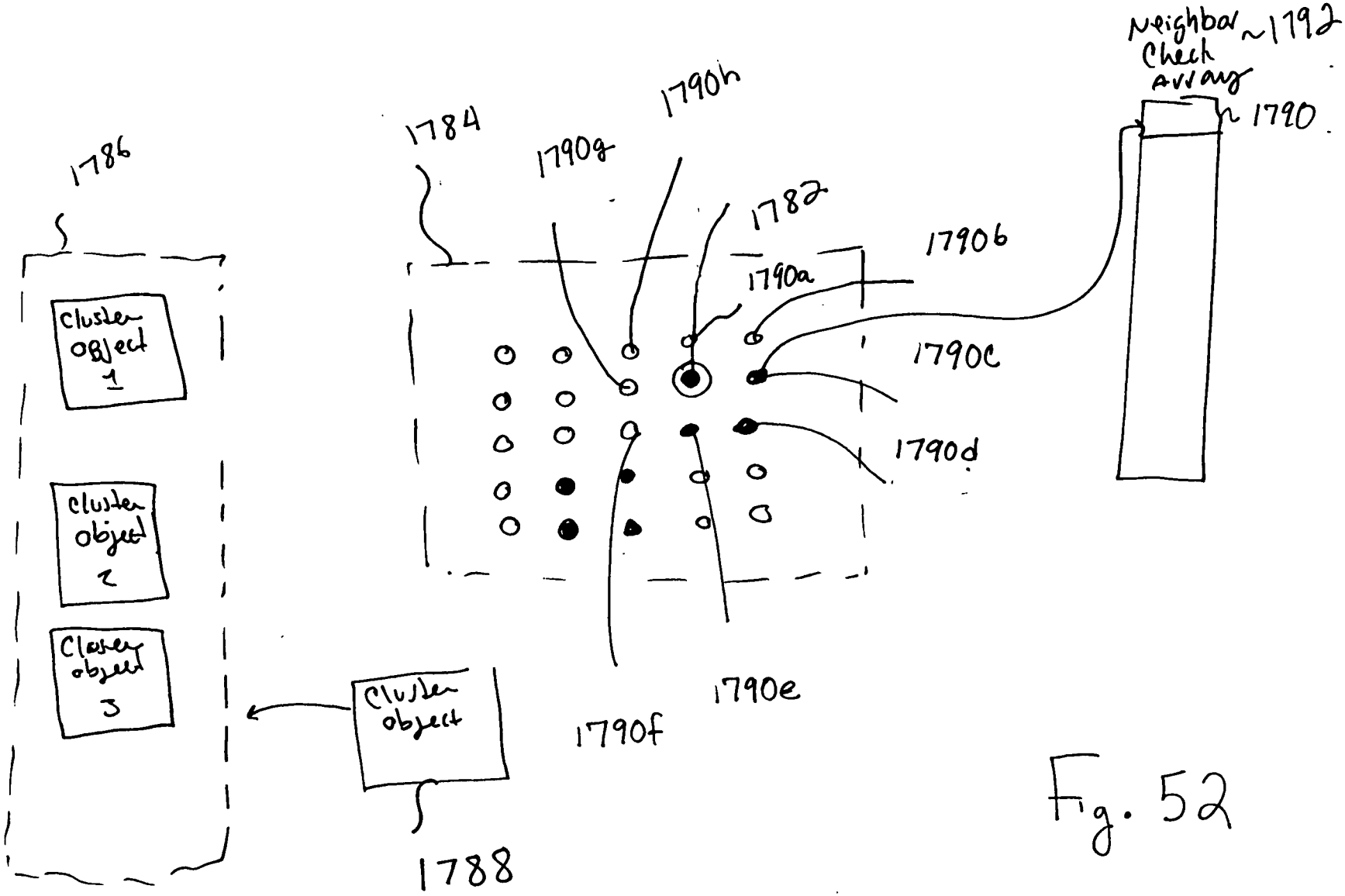


Fig. 52

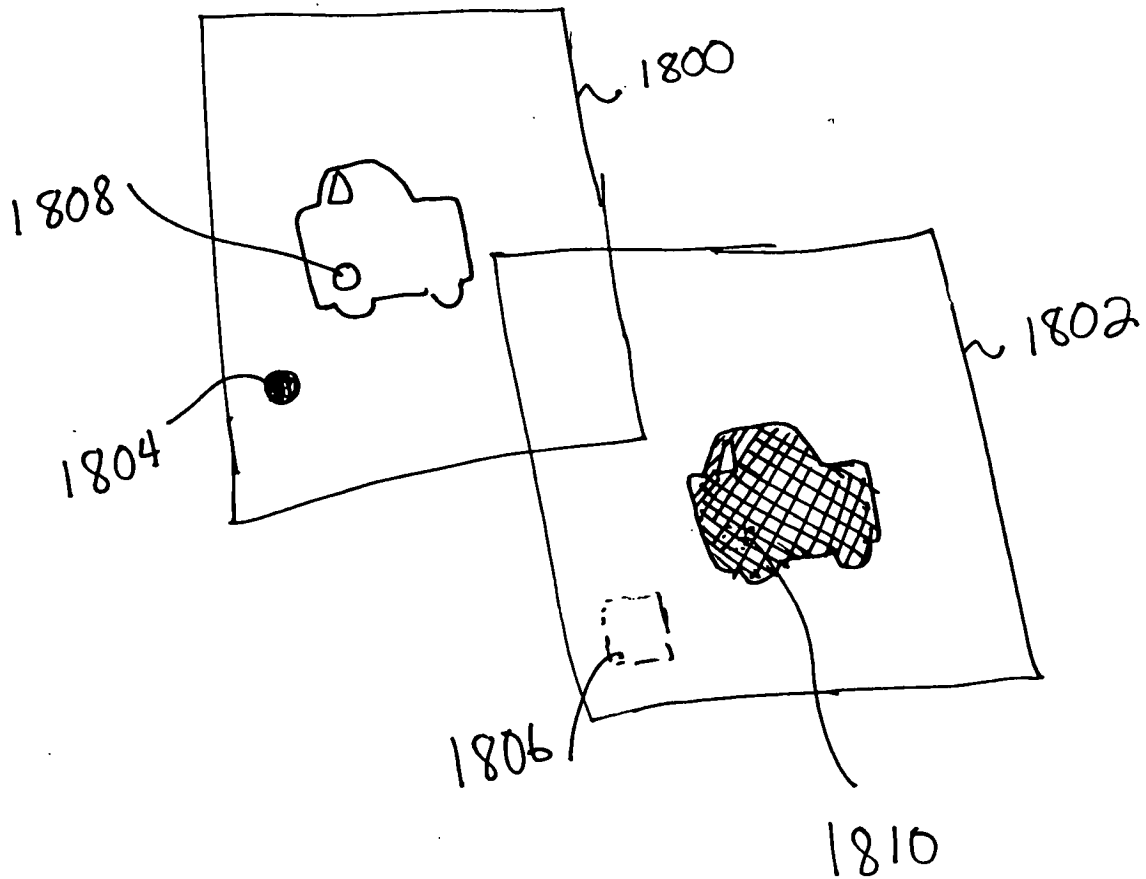


Fig. 53

Fig 54a

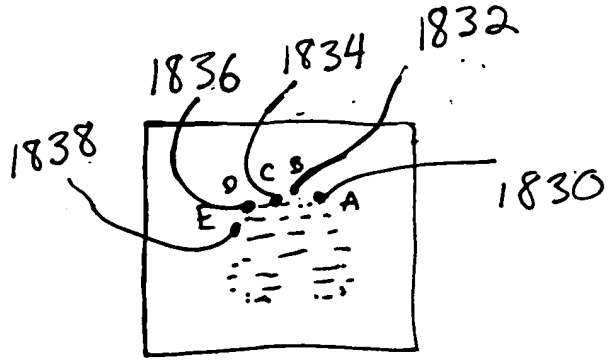


Fig. 54b

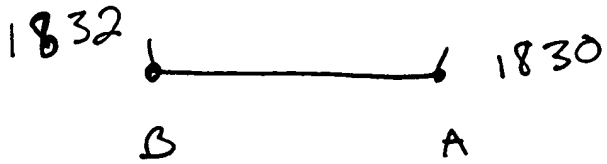


Fig. 54c

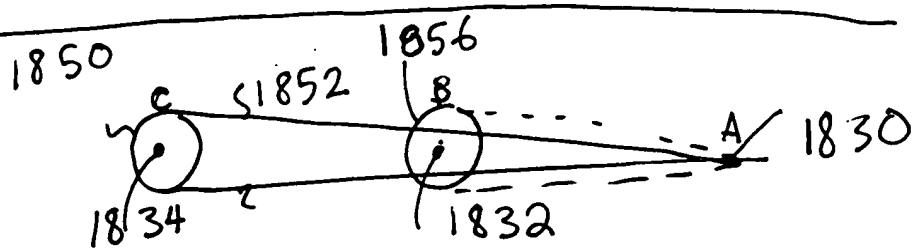


Fig. 54d

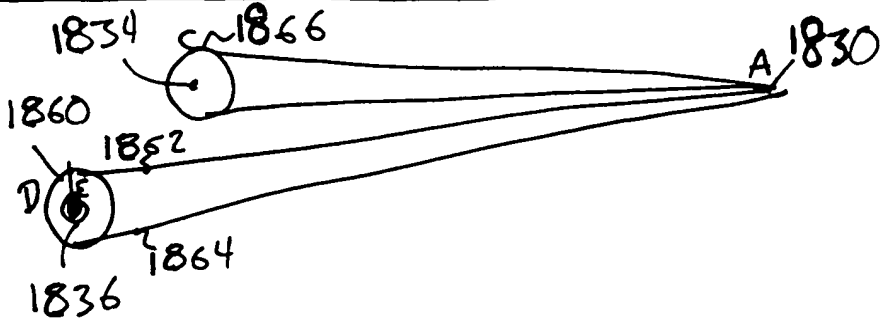
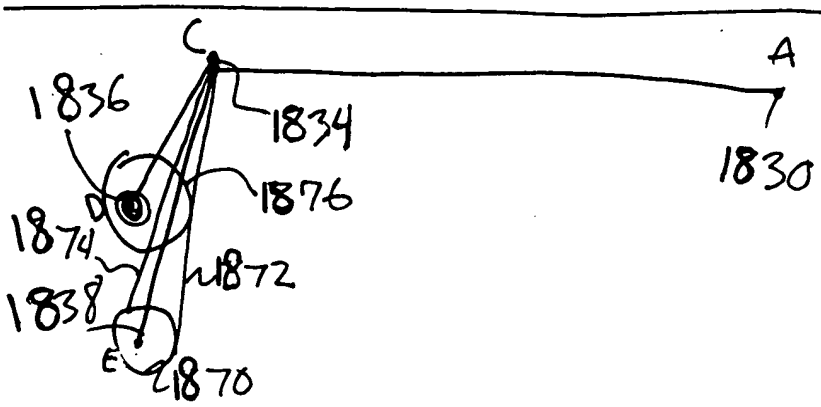


Fig. 55



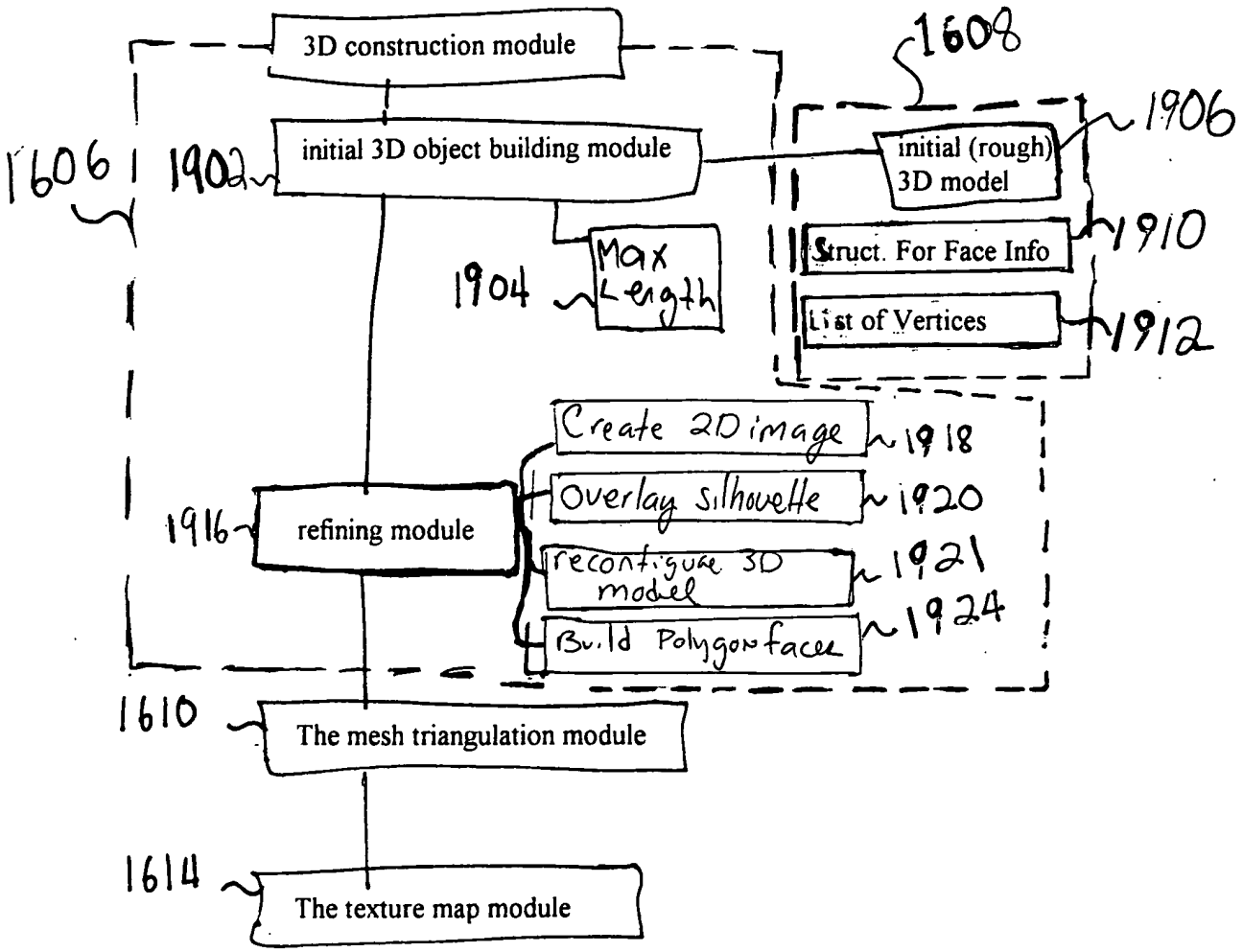

3D construction module
1608
1606
1902
initial 3D object building module
1906
initial (rough) 3D model
1904
Max Length
1910
Struct. For Face Info
List of Vertices
1912
Create 2D image
1918
Overlay silhouette
1920
1916
refining module
reconfigure 3D model
1921
Build Polygon faces
1924
1610
The mesh triangulation module
1614
The texture map module

Fig. 56

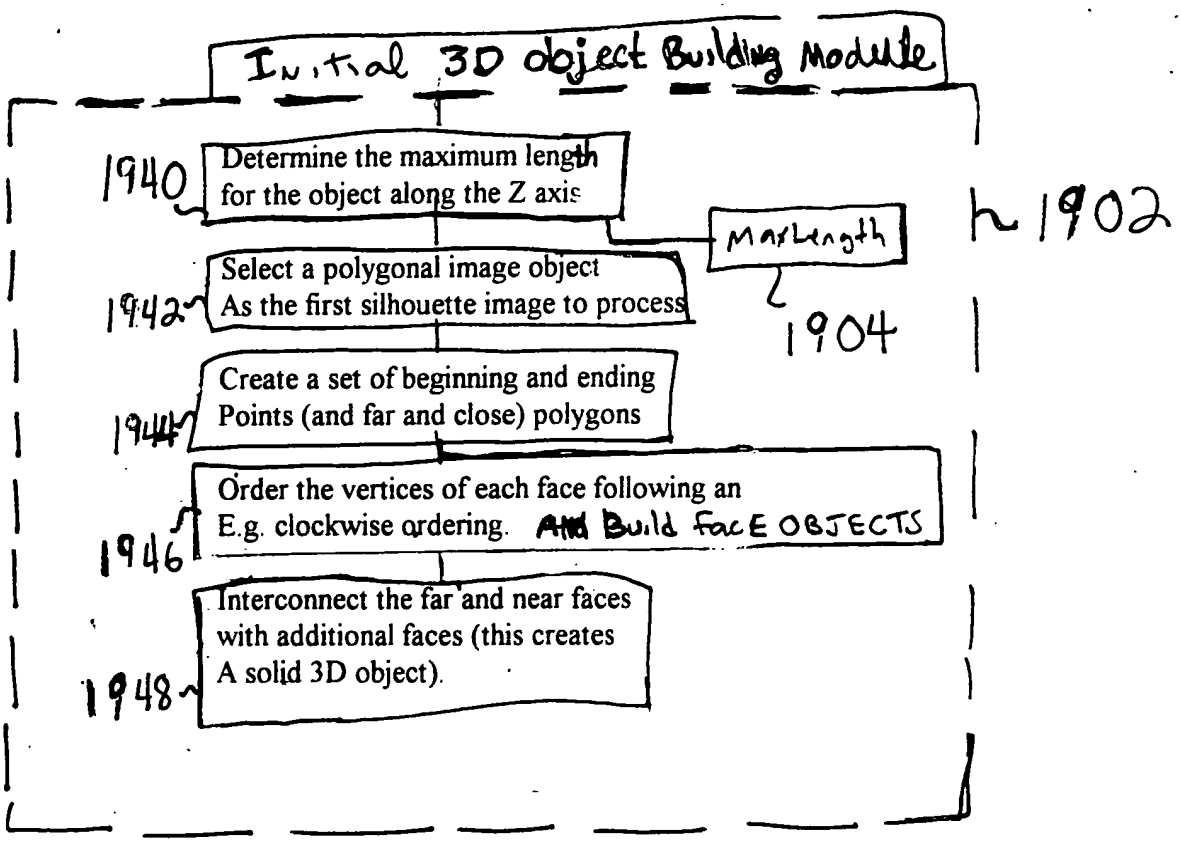

Initial 3D object Building Module
1940
Determine the maximum length for the object along the Z axis
MaxLength
1904
1902
1942
Select a polygonal image object As the first silhouette image to process
1944
Create a set of beginning and ending Points (and far and close) polygons
1946
Order the vertices of each face following an E.g. clockwise ordering. And Build FaceOBJECTS
1948
Interconnect the far and near faces with additional faces (this creates A solid 3D object).

Fig. 57

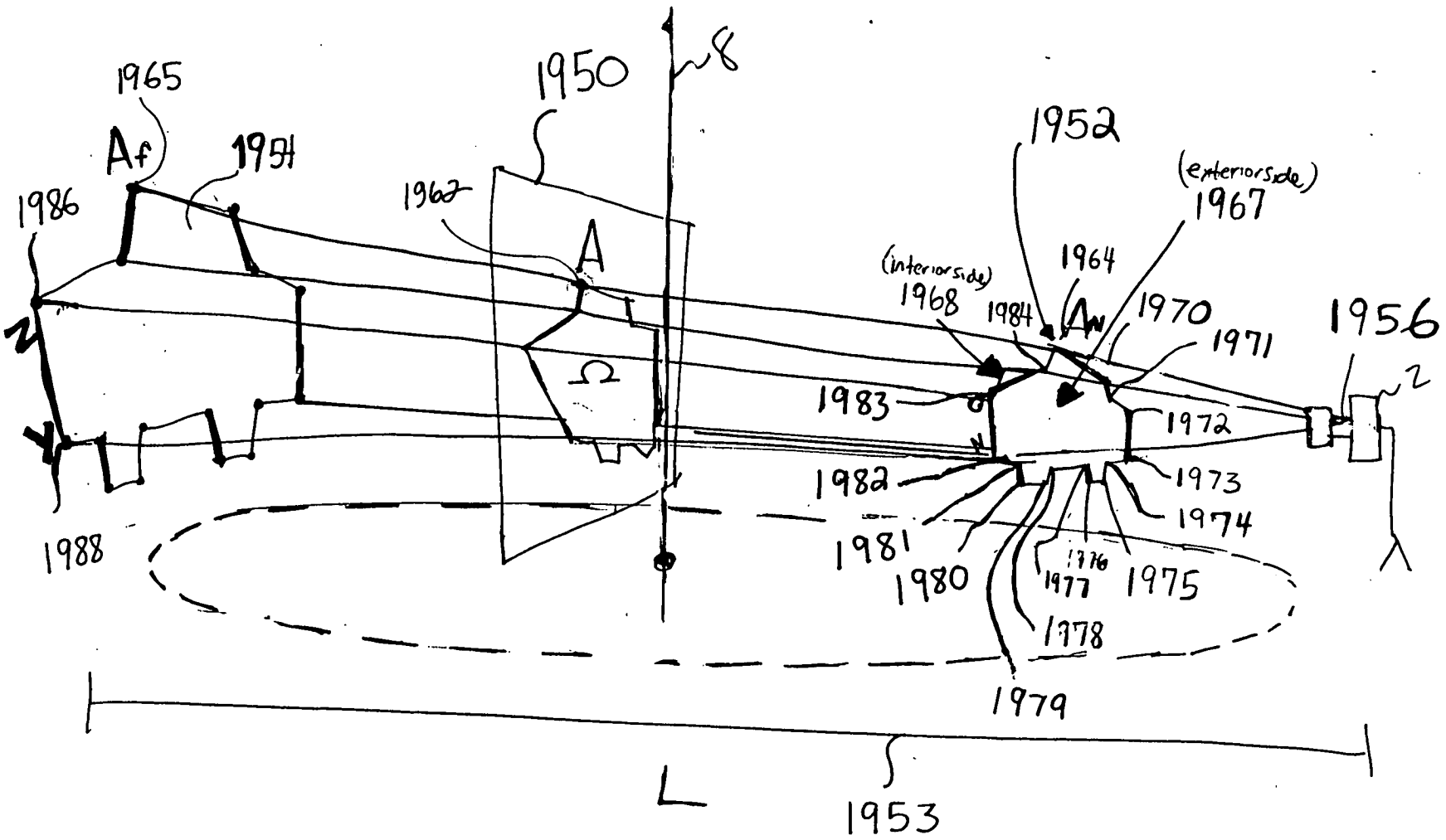


Fig. 58

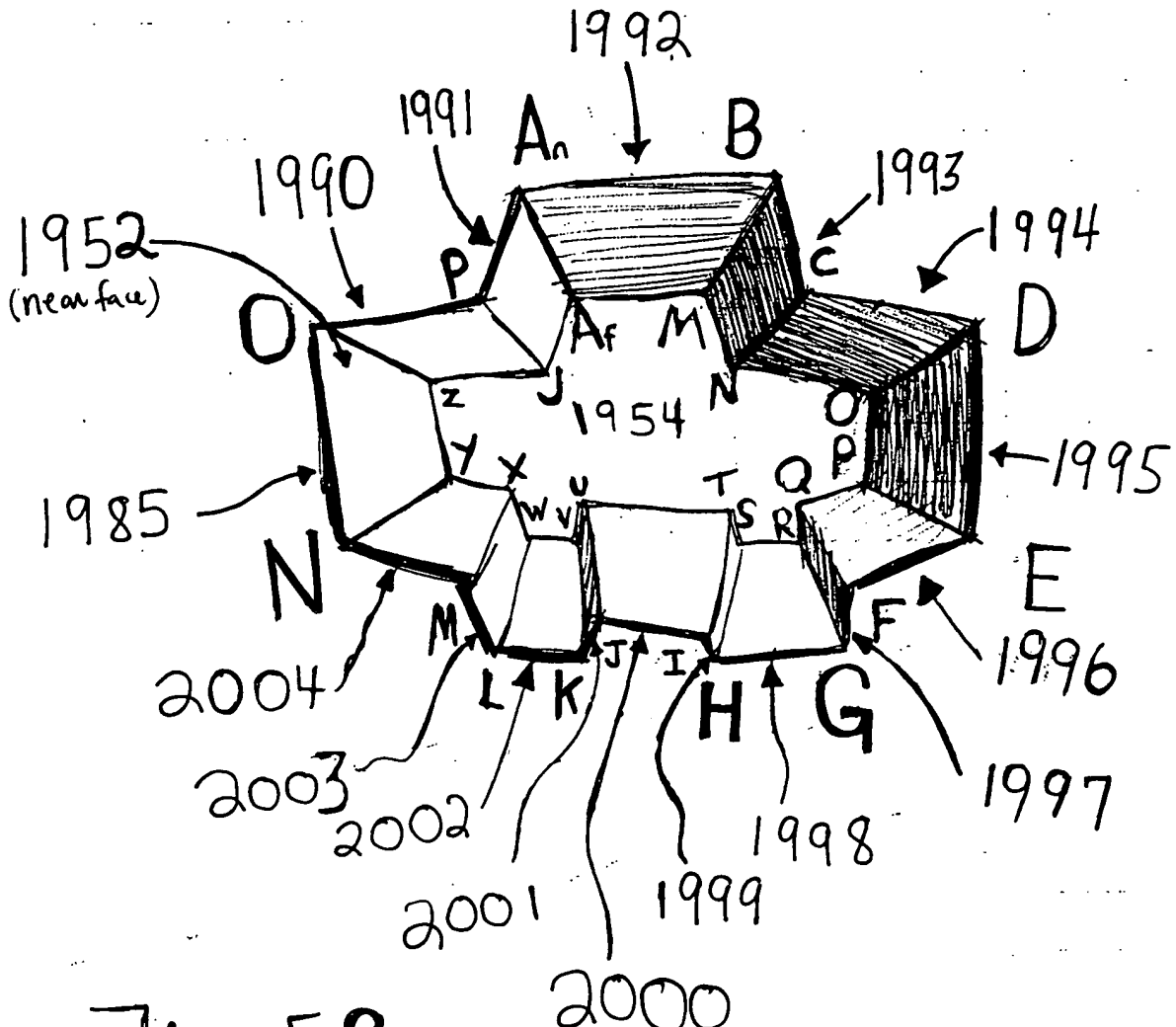

1992
1991
1990
1952
(near face)
1993
1994
1995
1985
2004
2003
2002
2001
2000
1999
1998
1997
1996
A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z
1954

Fig. 59

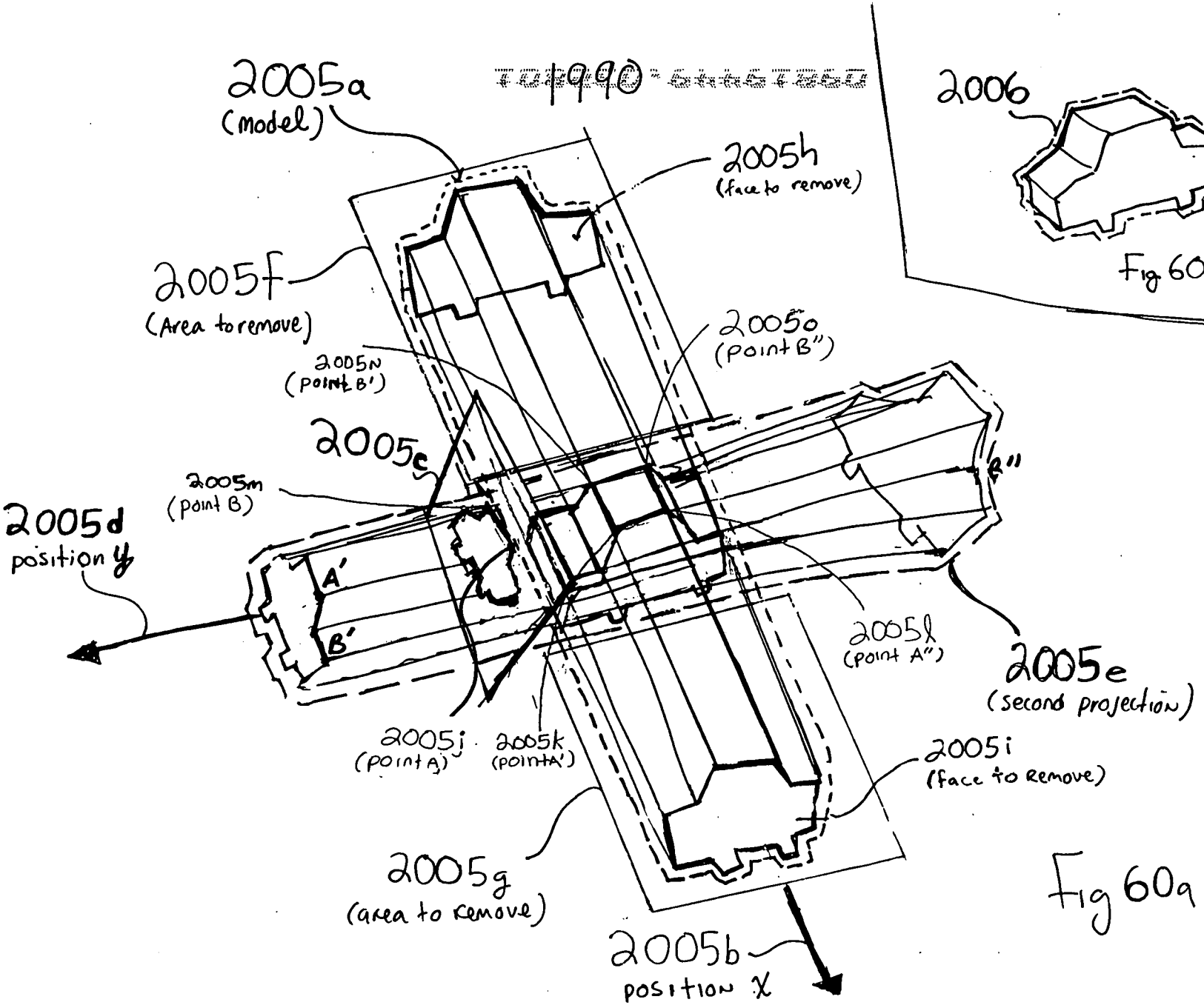


Fig 60a

Fig 60b

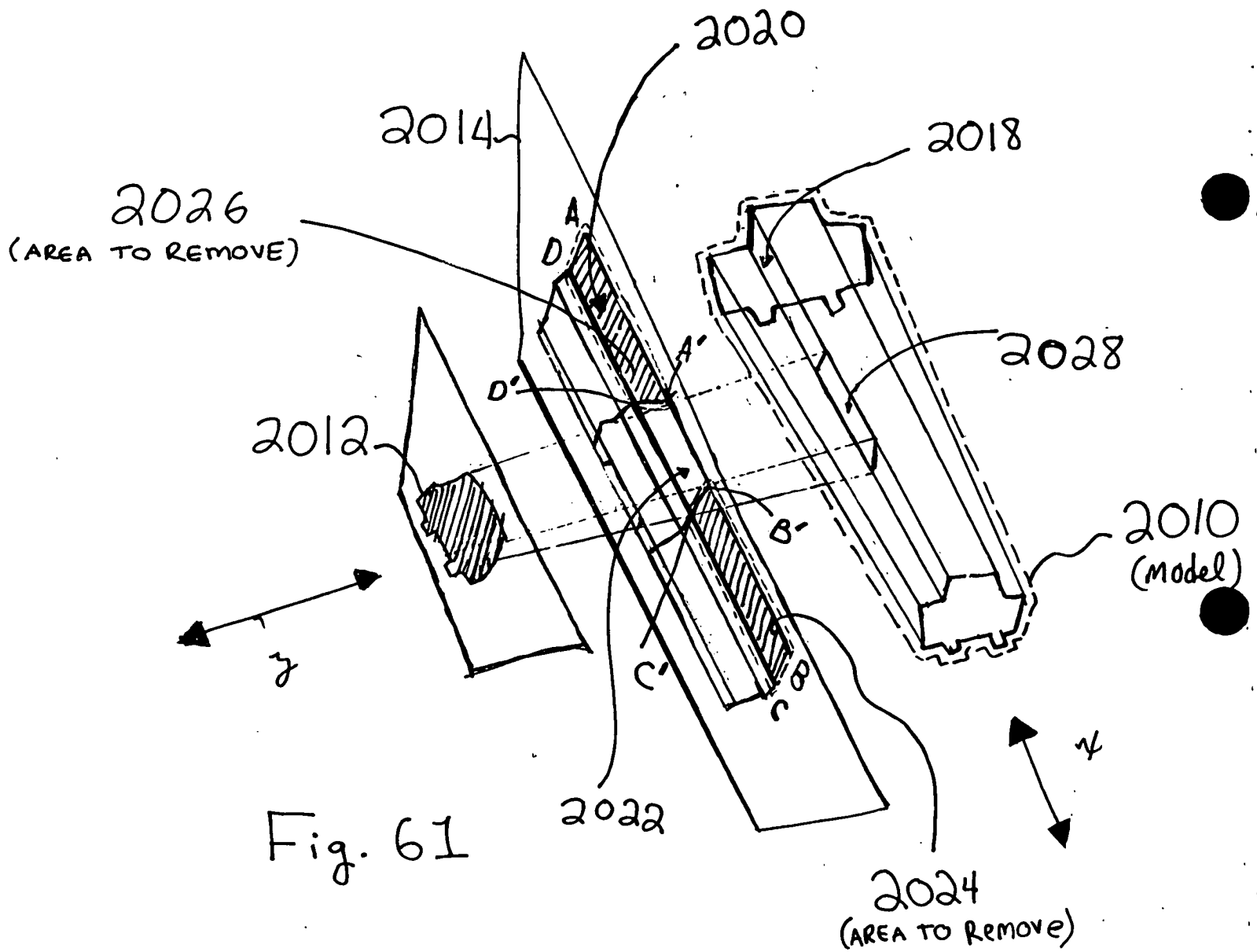


Fig. 61

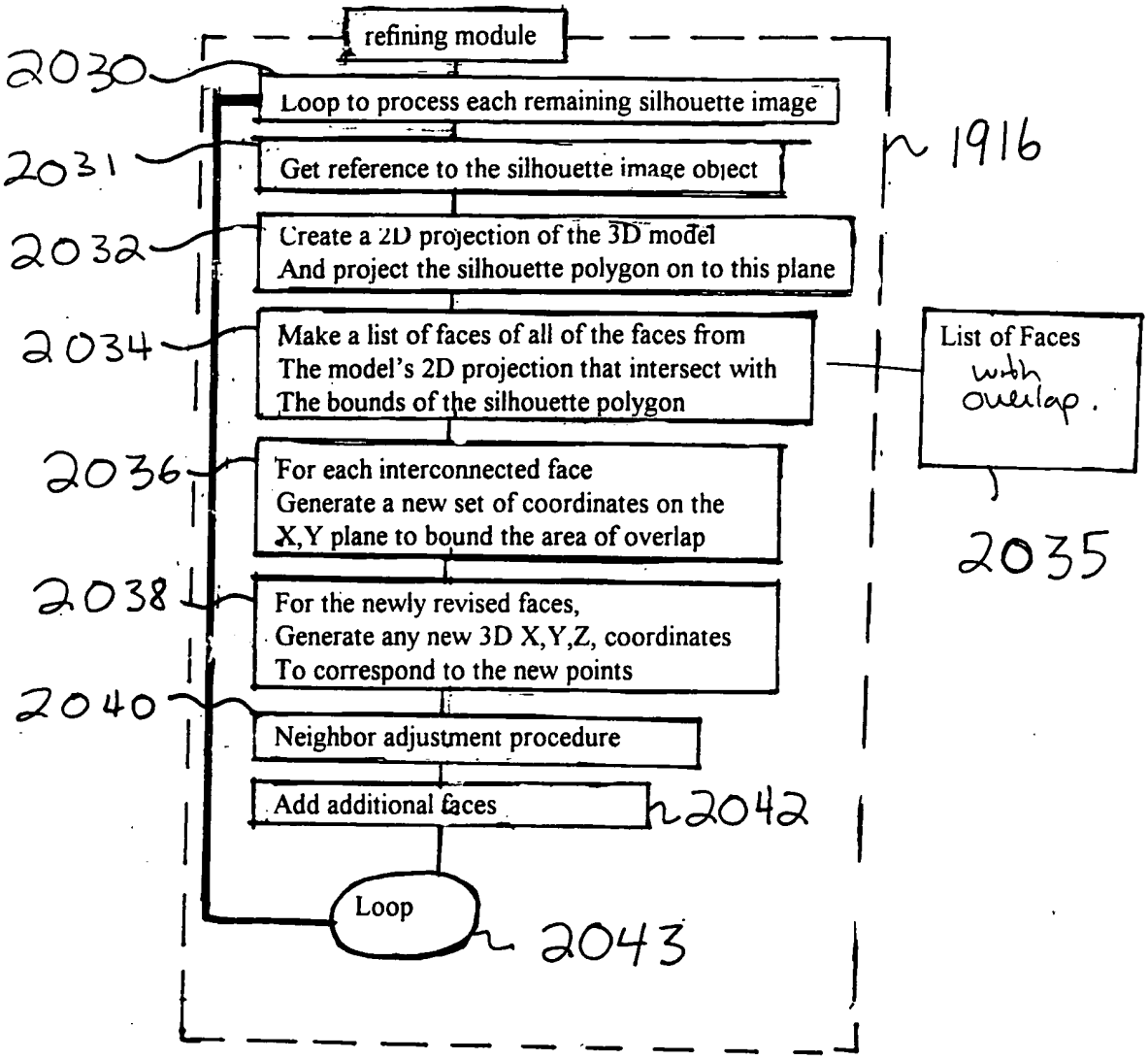

refining module
2030
Loop to process each remaining silhouette image
2031
Get reference to the silhouette image object
1916
2032
Create a 2D projection of the 3D model
And project the silhouette polygon on to this plane
2034
Make a list of faces of all of the faces from
The model's 2D projection that intersect with
The bounds of the silhouette polygon
List of Faces
with
overlap.
2036
For each interconnected face
Generate a new set of coordinates on the
X,Y plane to bound the area of overlap
2035
2038
For the newly revised faces,
Generate any new 3D X,Y,Z, coordinates
To correspond to the new points
2040
Neighbor adjustment procedure
Add additional faces
2042
Loop
2043

Fig. 62

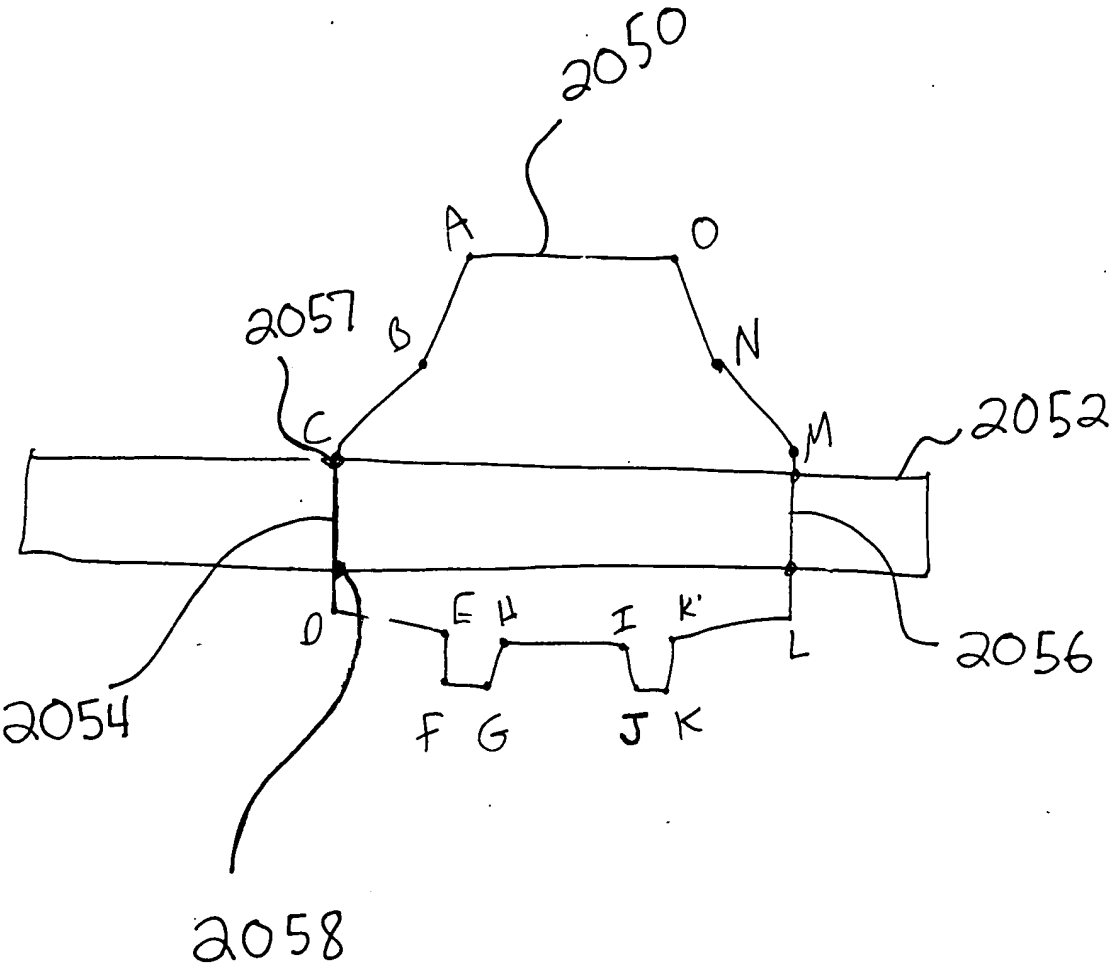


Fig. 63

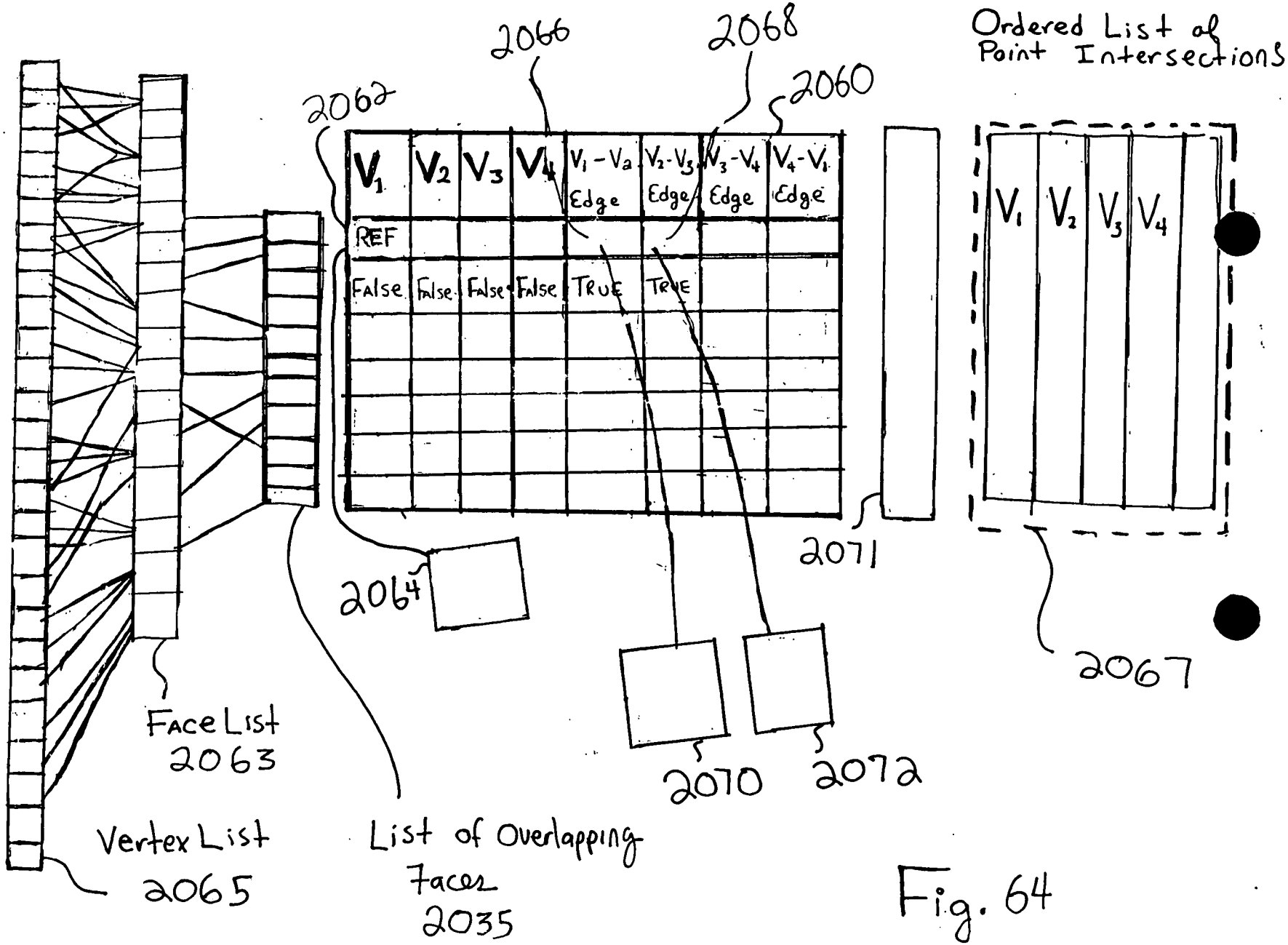


Fig. 64

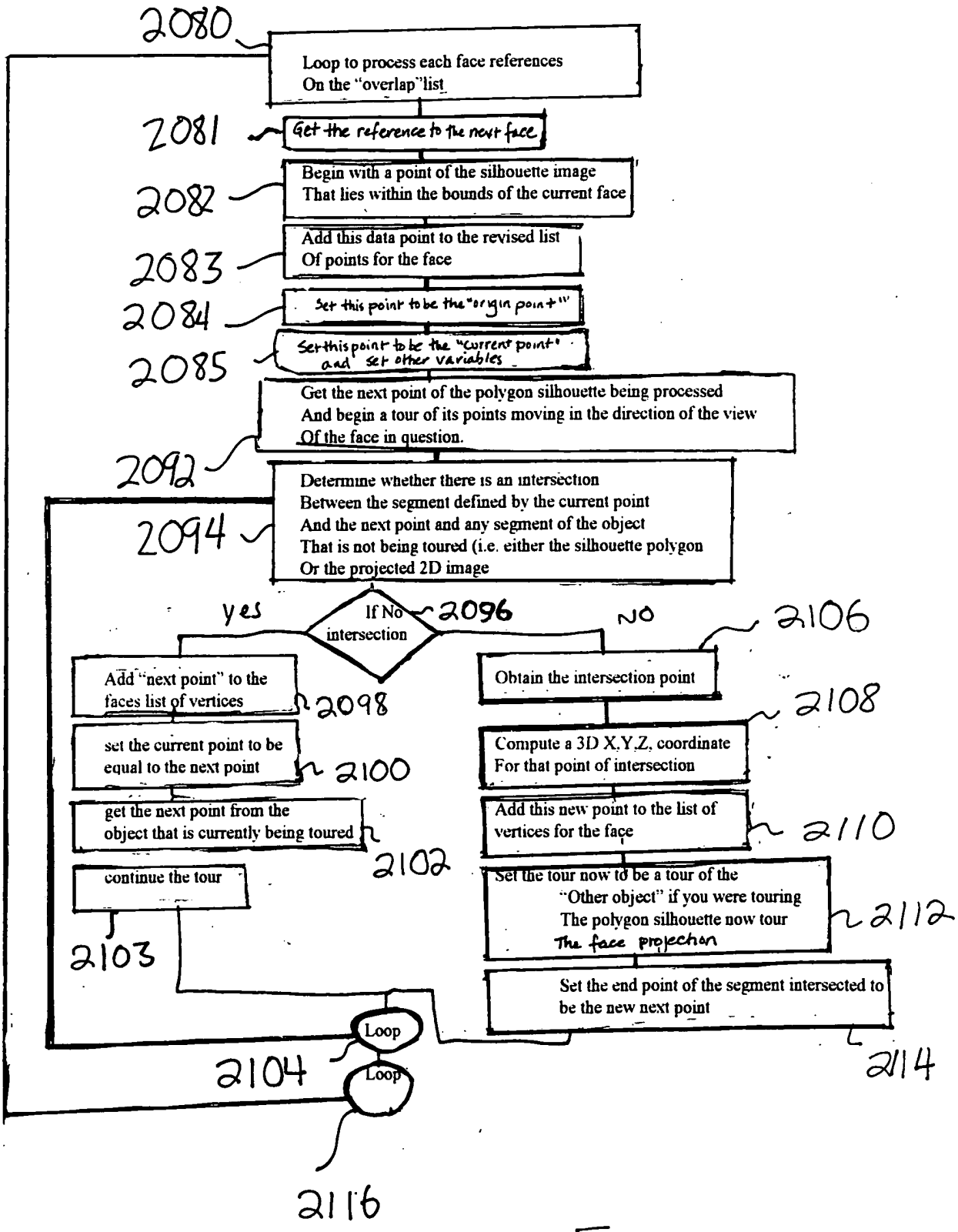

2080
Loop to process each face references On the "overlap" list
2081
Get the reference to the next face
2082
Begin with a point of the silhouette image That lies within the bounds of the current face
2083
Add this data point to the revised list Of points for the face
2084
Set this point to be the "origin point"
2085
Set this point to be the "current point" and set other variables
2092
Get the next point of the polygon silhouette being processed And begin a tour of its points moving in the direction of the view Of the face in question.
2094
Determine whether there is an intersection Between the segment defined by the current point And the next point and any segment of the object That is not being toured (i.e. either the silhouette polygon Or the projected 2D image
If No intersection
2096
yes
NO
2106
Add "next point" to the faces list of vertices
2098
Obtain the intersection point
2108
set the current point to be equal to the next point
2100
Compute a 3D X,Y,Z, coordinate For that point of intersection
get the next point from the object that is currently being toured
2102
Add this new point to the list of vertices for the face
2110
continue the tour
2103
Set the tour now to be a tour of the "Other object" if you were touring The polygon silhouette now tour The face projection
2112
Set the end point of the segment intersected to be the new next point
2114
Loop
2104
Loop
2116

Fig. 65

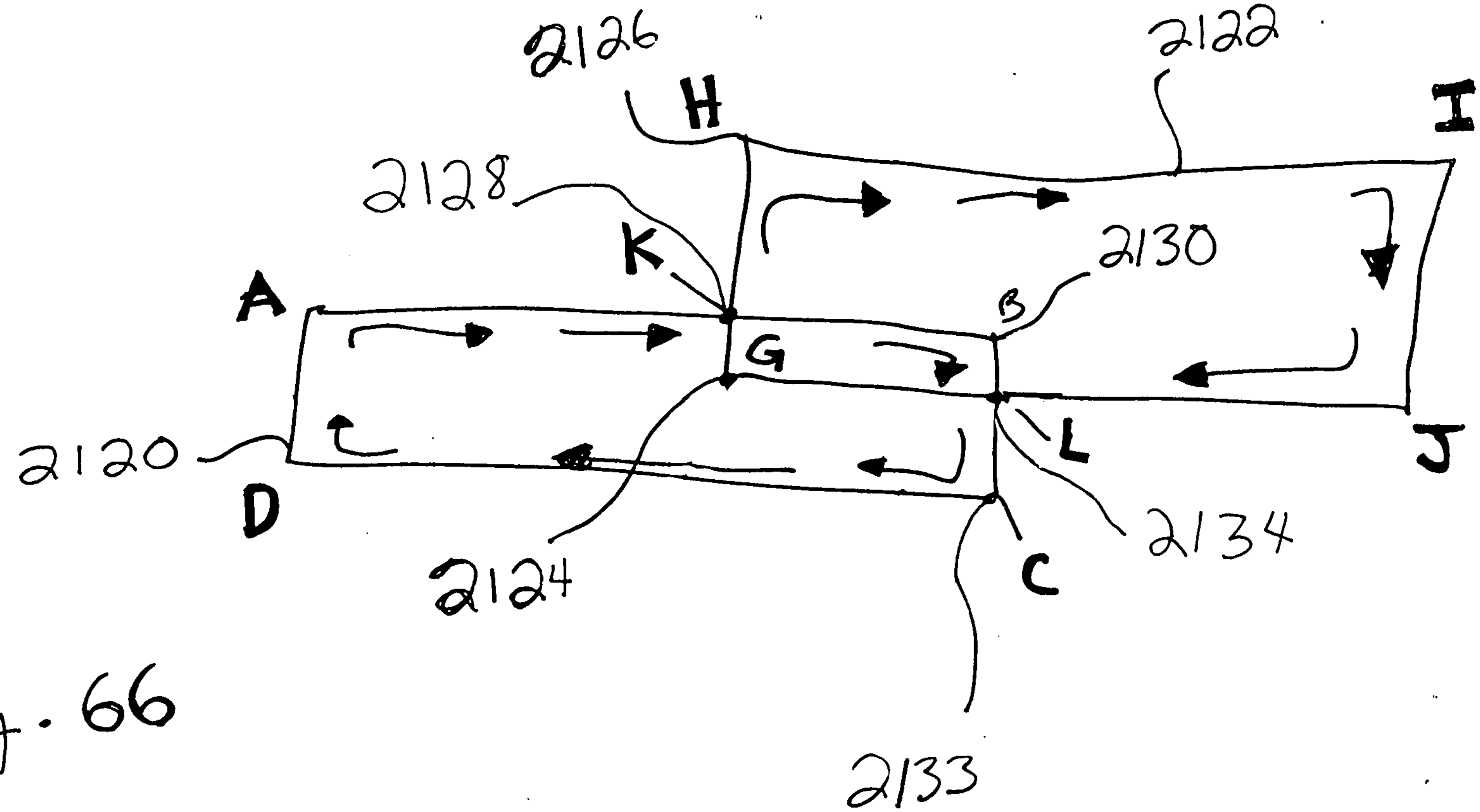


Fig. 66

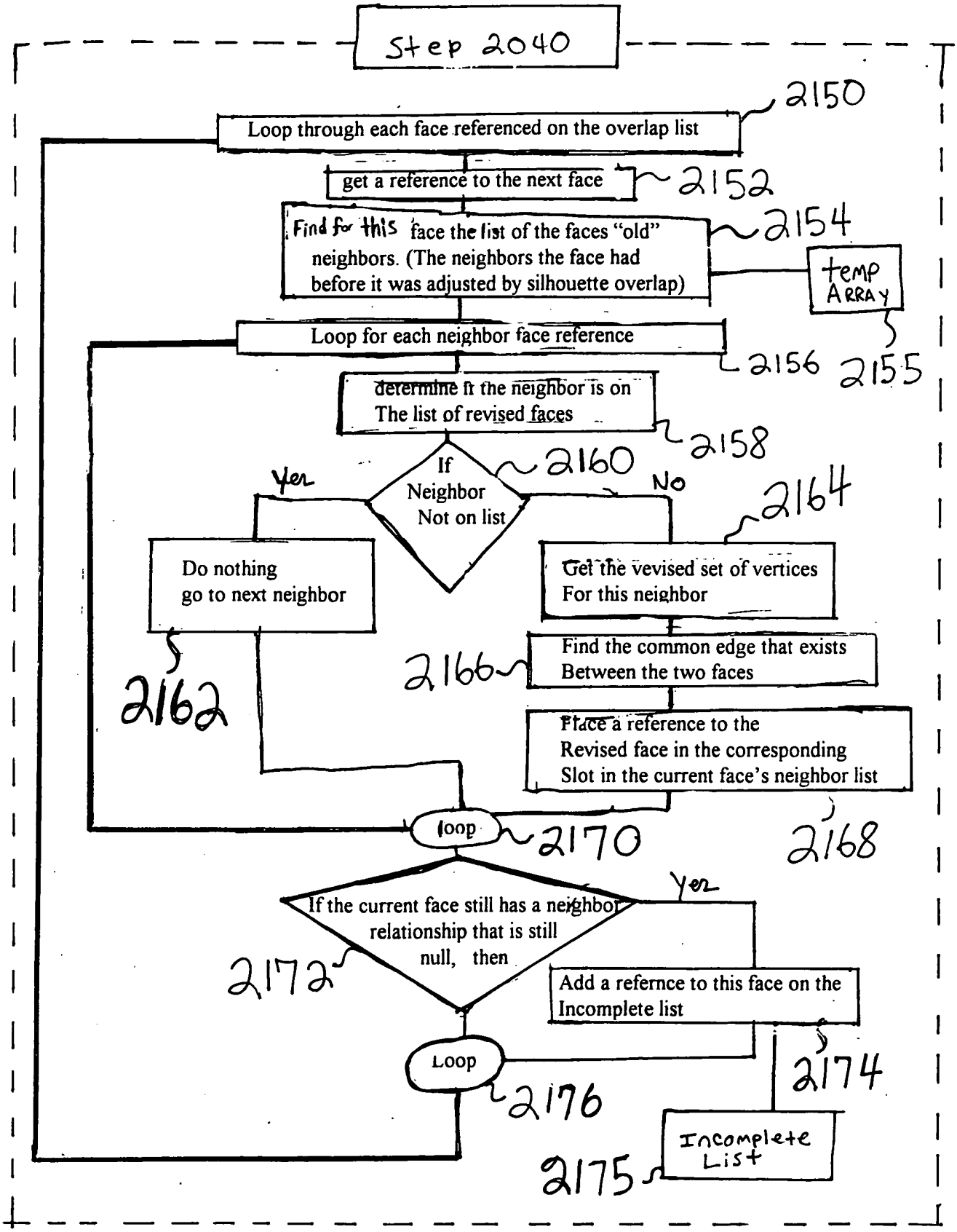

Step 2040
2150
Loop through each face referenced on the overlap list
get a reference to the next face
2152
Find for this face the list of the faces "old" neighbors. (The neighbors the face had before it was adjusted by silhouette overlap)
2154
temp ARRAY
2155
Loop for each neighbor face reference
2156
determine if the neighbor is on The list of revised faces
2158
If Neighbor Not on list
2160
Yes
No
2164
Do nothing go to next neighbor
2162
Get the revised set of vertices For this neighbor
2166
Find the common edge that exists Between the two faces
Place a reference to the Revised face in the corresponding Slot in the current face's neighbor list
2168
loop
2170
If the current face still has a neighbor relationship that is still null, then
2172
Yes
Add a refernce to this face on the Incomplete list
2174
Loop
2176
Incomplete List
2175

Fig. 67

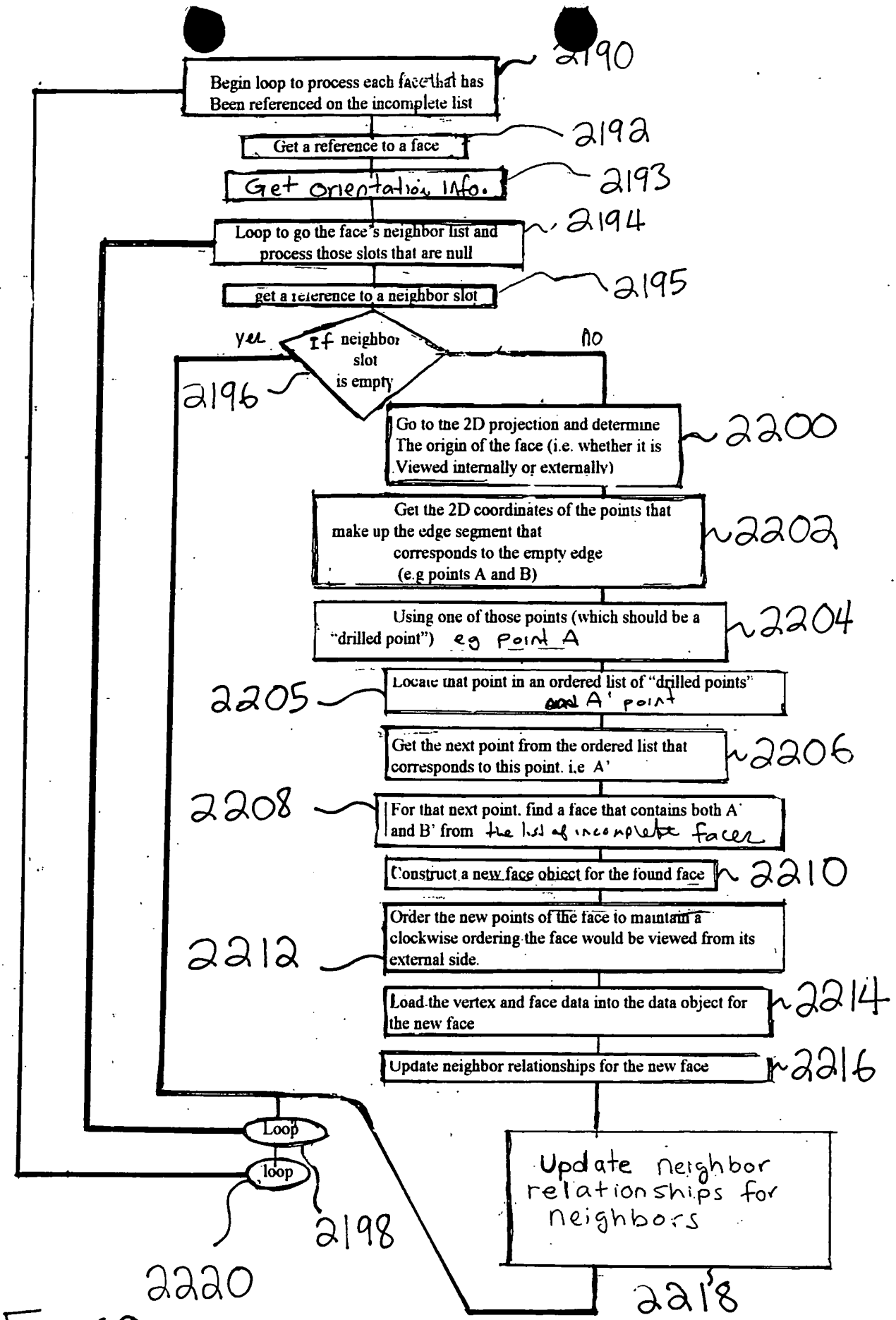

2190
Begin loop to process each face that has Been referenced on the incomplete list
2192
Get a reference to a face
2193
Get orientation info.
2194
Loop to go the face's neighbor list and process those slots that are null
2195
get a reference to a neighbor slot
yes
If neighbor slot is empty
no
2196
2200
Go to the 2D projection and determine The origin of the face (i.e. whether it is Viewed internally or externally)
2202
Get the 2D coordinates of the points that make up the edge segment that corresponds to the empty edge (e.g points A and B)
2204
Using one of those points (which should be a "drilled point") eg Point A
2205
Locate that point in an ordered list of "drilled points" and A' point
2206
Get the next point from the ordered list that corresponds to this point. i.e A'
2208
For that next point. find a face that contains both A' and B' from the list of incomplete faces
2210
Construct a new face object for the found face
2212
Order the new points of the face to maintain a clockwise ordering the face would be viewed from its external side.
2214
Load the vertex and face data into the data object for the new face
2216
Update neighbor relationships for the new face
Loop
loop
2198
2220
Update neighbor relationships for neighbors
2218

Fig. 68

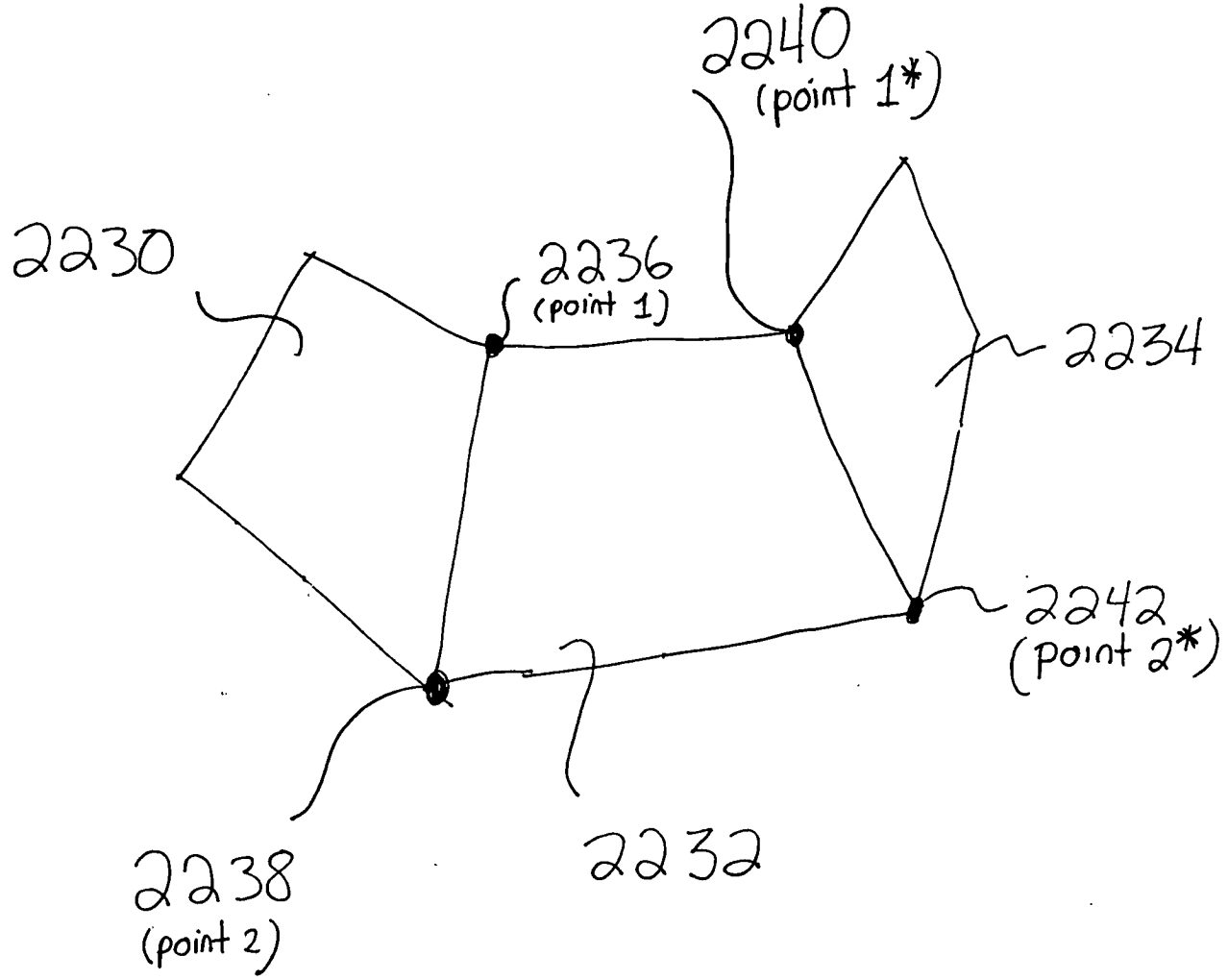


Fig. 69